

C MCQs on Data Types, Operators and Expressions

Variable Names – 1

1. C99 standard guarantees uniqueness of _____ characters for internal names.

- a) 31
- b) 63
- c) 12
- d) 14

[View Answer](#)

Answer: b

Explanation: ISO C99 compiler may consider only first 63 characters for internal names.

2. C99 standard guarantees uniqueness of _____ characters for external names.

- a) 31
- b) 6
- c) 12
- d) 14

[View Answer](#)

Answer: a

Explanation: ISO C99 compiler may consider only first 31 characters for external names.

3. Which of the following is not a valid variable name declaration?

- a) int __a3;
- b) int __3a;
- c) int __A3;
- d) None of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

4. Which of the following is not a valid variable name declaration?

- a) int _a3;
- b) int a_3;
- c) int 3_a;
- d) int _3a;

[View Answer](#)

Answer: c

Explanation: Variable name cannot start with a digit.

5. Why do variable names beginning with the underscore is not encouraged?

- a) It is not standardized
- b) To avoid conflicts since assemblers and loaders use such names
- c) To avoid conflicts since library routines use such names
- d) To avoid conflicts with environment variables of an operating system

[View Answer](#)

Answer: c

Explanation: None.

6. All keywords in C are in _____

- a) LowerCase letters
- b) UpperCase letters
- c) CamelCase letters
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

7. Variable name resolution (number of significant characters for the uniqueness of variable) depends on _____

- a) Compiler and linker implementations
- b) Assemblers and loaders implementations
- c) C language
- d) None of the mentioned

View Answer

Answer: a

Explanation: It depends on the standard to which compiler and linkers are adhering.

8. Which of the following is not a valid C variable name?

- a) int number;
- b) float rate;
- c) int variable_count;
- d) int \$main;

View Answer

Answer: d

Explanation: Since only underscore and no other special character is allowed in a variable name, it results in an error.

9. Which of the following is true for variable names in C?

- a) They can contain alphanumeric characters as well as special characters
- b) It is not an error to declare a variable to be one of the keywords (like goto, static)
- c) Variable names cannot start with a digit
- d) Variable can be of any length

View Answer

Answer: c

Explanation: According to the syntax for C variable name, it cannot start with a digit.

Variable Names – 2

1. Which is valid C expression?

- a) int my_num = 100,000;
- b) int my_num = 100000;
- c) int my num = 1000;

d) `int $my_num = 10000;`

[View Answer](#)

Answer: b

Explanation: Space, comma and \$ cannot be used in a variable name.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        printf("Hello World! %d \n", x);
5.        return 0;
6.    }
```

- a) Hello World! x;
- b) Hello World! followed by a junk value
- c) Compile time error
- d) Hello World!

[View Answer](#)

Answer: c

Explanation: It results in an error since x is used without declaring the variable x.

Output:

```
$ cc pgm1.c
```

```
pgm1.c: In function 'main':
```

```
pgm1.c:4: error: 'x' undeclared (first use in this function)
```

```
pgm1.c:4: error: (Each undeclared identifier is reported only once
```

```
pgm1.c:4: error: for each function it appears in.)
```

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int y = 10000;
5.        int y = 34;
6.        printf("Hello World! %d\n", y);
7.        return 0;
8.    }
```

- a) Compile time error
- b) Hello World! 34
- c) Hello World! 1000

d) Hello World! followed by a junk value

[View Answer](#)

Answer: a

Explanation: Since y is already defined, redefining it results in an error.

Output:

```
$ cc pgm2.c
```

```
pgm2.c: In function 'main':
```

```
pgm2.c:5: error: redefinition of 'y'
```

```
pgm2.c:4: note: previous definition of 'y' was here
```

4. Which of the following is not a valid variable name declaration?

a) float PI = 3.14;

b) double PI = 3.14;

c) int PI = 3.14;

d) #define PI 3.14

[View Answer](#)

Answer: d

Explanation: #define PI 3.14 is a macro preprocessor, it is a textual substitution.

5. What will happen if the following C code is executed?

```
1.    #include <stdio.h>
```

```
2.    int main()
```

```
3.    {
```

```
4.        int main = 3;
```

```
5.        printf("%d", main);
```

```
6.        return 0;
```

```
7.    }
```

a) It will cause a compile-time error

b) It will cause a run-time error

c) It will run without any error and prints 3

d) It will experience infinite looping

[View Answer](#)

Answer: c

Explanation: A C program can have same function name and same variable name.

```
$ cc pgm3.c
```

```
$ a.out
```

```
3
```

6. What is the problem in the following variable declaration?

```
float 3Bedroom-Hall-Kitchen?;
```

a) The variable name begins with an integer

b) The special character '-'

- c) The special character '?'
- d) All of the mentioned

View Answer

Answer: d

Explanation: A variable name cannot start with an integer, along with that the C compiler interprets the '-' and '?' as a minus operator and a question mark operator respectively.

7. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.
3.    int main()
4.    {
5.        int ThisIsVariableName = 12;
6.        int ThisIsVariablename = 14;
7.        printf("%d", ThisIsVariablename);
8.        return 0;
9.    }
```

- a) The program will print 12
- b) The program will print 14
- c) The program will have a runtime error
- d) The program will cause a compile-time error due to redeclaration

View Answer

Answer: b

Explanation: Variable names ThisIsVariablename and ThisIsVariableName are both distinct as C is case sensitive.

Output:

```
$ cc pgm4.c
```

```
$ a.out
```

```
14
```

8. Which of the following cannot be a variable name in C?

- a) volatile
- b) true
- c) friend
- d) export

View Answer

Answer: a

Explanation: volatile is C keyword.

Data Types and Sizes – 1

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```

2.   int main()
3.   {
4.       int a[5] = {1, 2, 3, 4, 5};
5.       int i;
6.       for (i = 0; i < 5; i++)
7.           if ((char)a[i] == '5')
8.               printf("%d\n", a[i]);
9.           else
10.              printf("FAIL\n");
11.  }

```

- a) The compiler will flag an error
- b) The program will compile and print the output 5
- c) The program will compile and print the ASCII value of 5
- d) The program will compile and print FAIL for 5 times

View Answer

Answer: d

Explanation: The ASCII value of 5 is 53, the char type-casted integral value 5 is 5 only.

Output:

\$ cc pgm1.c

\$ a.out

FAIL

FAIL

FAIL

FAIL

FAIL

2. The format identifier '%i' is also used for _____ data type.

- a) char
- b) int
- c) float
- d) double

View Answer

Answer: b

Explanation: Both %d and %i can be used as a format identifier for int data type.

3. Which data type is most suitable for storing a number 65000 in a 32-bit system?

- a) signed short
- b) unsigned short
- c) long
- d) int

View Answer

Answer: b

Explanation: 65000 comes in the range of short (16-bit) which occupies the least memory. Signed short ranges from -32768 to 32767 and hence we should use unsigned short.

4. Which of the following is a User-defined data type?

- a) typedef int Boolean;
- b) typedef enum {Mon, Tue, Wed, Thu, Fri} Workdays;
- c) struct {char name[10], int age};
- d) all of the mentioned

View Answer

Answer: d

Explanation: typedef and struct are used to define user-defined data types.

5. What is the size of an int data type?

- a) 4 Bytes
- b) 8 Bytes
- c) Depends on the system/compiler
- d) Cannot be determined

View Answer

Answer: c

Explanation: The size of the data types depend on the system.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      signed char chr;
5.      chr = 128;
6.      printf("%d\n", chr);
7.      return 0;
8.  }
```

- a) 128
- b) -128
- c) Depends on the compiler
- d) None of the mentioned

View Answer

Answer: b

Explanation: The range of signed character is from -128 to +127. Since we are assigning a value of 128 to the variable 'chr', the result will be negative. 128 in binary is represented as "1000 0000" for character datatype. As you can see that the sign bit is set to 1, followed by 7 zeros (0), its final decimal value will be -128 (negative 128).

Output:

```
$ cc pgm2.c
$ a.out
-128
```

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char c;
5.      int i = 0;
6.      FILE *file;
7.      file = fopen("test.txt", "w+");
8.      fprintf(file, "%c", 'a');
9.      fprintf(file, "%c", -1);
10.     fprintf(file, "%c", 'b');
11.     fclose(file);
12.     file = fopen("test.txt", "r");
13.     while ((c = fgetc(file)) != -1)
14.         printf("%c", c);
15.     return 0;
16. }
```

- a) a
- b) Infinite loop
- c) Depends on what fgetc returns
- d) Depends on the compiler

View Answer

Answer: a

Explanation: None.

Output:

```
$ cc pgm3.c
$ a.out
a
```

8. What is short int in C programming?

- a) The basic data type of C
- b) Qualifier
- c) Short is the qualifier and int is the basic data type
- d) All of the mentioned

View Answer

Answer: c

Explanation: None.

Data Types and Sizes – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        float f1 = 0.1;
5.        if (f1 == 0.1)
6.            printf("equal\n");
7.        else
8.            printf("not equal\n");
9.    }
```

- a) equal
- b) not equal
- c) output depends on the compiler
- d) error

View Answer

Answer: b

Explanation: 0.1 by default is of type double which has different representation than float resulting in inequality even after conversion.

Output:

```
$ cc pgm4.c
```

```
$ a.out
```

```
not equal
```

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        float f1 = 0.1;
5.        if (f1 == 0.1f)
6.            printf("equal\n");
7.        else
8.            printf("not equal\n");
9.    }
```

- a) equal
- b) not equal
- c) output depends on compiler
- d) error

View Answer

Answer: a

Explanation: 0.1f results in 0.1 to be stored in floating point representations.

Output:

```
$ cc pgm5.c
```

```
$ a.out
```

```
equal
```

3. What will be the output of the following C code on a 32-bit machine?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 10000;
5.      double y = 56;
6.      int *p = &x;
7.      double *q = &y;
8.      printf("p and q are %d and %d", sizeof(p), sizeof(q));
9.      return 0;
10. }
```

- a) p and q are 4 and 4
- b) p and q are 4 and 8
- c) compiler error
- d) p and q are 2 and 8

View Answer

Answer: a

Explanation: Size of any type of pointer is 4 on a 32-bit machine.

Output:

```
$ cc pgm6.c
```

```
$ a.out
```

```
p and q are 4 and 4
```

4. Which is correct with respect to the size of the data types?

- a) char > int > float
- b) int > char > float
- c) char < int < double
- d) double > char > int

View Answer

Answer: c

Explanation: char has less bytes than int and int has less bytes than double in any system

5. What will be the output of the following C code on a 64 bit machine?

```
1.  #include <stdio.h>
2.  union Sti
3.  {
4.      int nu;
5.      char m;
6.  };
7.  int main()
8.  {
9.      union Sti s;
10.     printf("%d", sizeof(s));
11.     return 0;
12. }
```

- a) 8
- b) 5
- c) 9
- d) 4

View Answer

Answer: d

Explanation: Since the size of a union is the size of its maximum data type, here int is the largest data type. Hence the size of the union is 4.

Output:

```
$ cc pgm7.c
```

```
$ a.out
```

```
4
```

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      float x = 'a';
5.      printf("%f", x);
6.      return 0;
7.  }
```

- a) a
- b) run time error
- c) a.0000000
- d) 97.000000

View Answer

Answer: d

Explanation: Since the ASCII value of a is 97, the same is assigned to the float variable and printed.

Output:

```
$ cc pgm8.c
$ a.out
97.000000
```

7. Which of the data types has the size that is variable?

- a) int
- b) struct
- c) float
- d) double

View Answer

Answer: b

Explanation: Since the size of the structure depends on its fields, it has a variable size.

Constants – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      enum {ORANGE = 5, MANGO, BANANA = 4, PEACH};
5.      printf("PEACH = %d\n", PEACH);
6.  }
```

- a) PEACH = 3
- b) PEACH = 4
- c) PEACH = 5
- d) PEACH = 6

View Answer

Answer: c

Explanation: In enum, the value of constant is defined to the recent assignment from left.

Output:

```
$ cc pgm1.c
$ a.out
PEACH = 5
```

2. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        printf("C programming %s", "Class by\n%s Sanfoundry", "WOW");
5.    }

```

a)

C programming Class by

WOW Sanfoundry

b) C programming Class by\n%s Sanfoundry

c)

C programming Class by

%s Sanfoundry

d) Compilation error

View Answer

Answer: c

Explanation: This program has only one %s within first double quotes, so it does not read the string "WOW".

The %s along with the Sanfoundry is not read as a format modifier while new line character prints the new line.

Output:

\$ cc pgm2.c

\$ a.out

C programming Class by

%s Sanfoundry

3. In the following code snippet, character pointer str holds a reference to the string _____

```
char *str = "Sanfoundry.com\0" "training classes";
```

a) Sanfoundry.com

b) Sanfoundry.com\0training classes

c) Sanfoundry.comtraining classes

d) Invalid declaration

View Answer

Answer: b

Explanation: '\0' is accepted as a char in the string. Even though strlen will give length of string "Sanfoundry.com", in memory str is pointing to entire string including training classes.

4. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    #define a 10

```

```

3.   int main()
4.   {
5.       const int a = 5;
6.       printf("a = %d\n", a);
7.   }

```

- a) a = 5
- b) a = 10
- c) Compilation error
- d) Runtime error

View Answer

Answer: c

Explanation: The #define substitutes a with 10 without leaving any identifier, which results in Compilation error.

Output:

```
$ cc pgm3.c
```

```
pgm3.c: In function 'main':
```

```
pgm3.c:5: error: expected identifier or '(' before numeric constant
```

5. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int main()
3.   {
4.       int var = 010;
5.       printf("%d", var);
6.   }

```

- a) 2
- b) 8
- c) 9
- d) 10

View Answer

Answer: b

Explanation: 010 is octal representation of 8.

Output:

```
$ cc pgm4.c
```

```
$ a.out
```

```
8
```

6. What will be the output of the following C function?

```

1.   #include <stdio.h>
2.   enum birds {SPARROW, PEACOCK, PARROT};

```

```

3.    enum animals {TIGER = 8, LION, RABBIT, ZEBRA};
4.    int main()
5.    {
6.        enum birds m = TIGER;
7.        int k;
8.        k = m;
9.        printf("%d\n", k);
10.     return 0;
11. }

```

- a) 0
- b) Compile time error
- c) 1
- d) 8

View Answer

Answer: d

Explanation: m is an integer constant, hence it is compatible.

Output:

```
$ cc pgm5.c
```

```
$ a.out
```

```
8
```

7. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    #define MAX 2
3.    enum bird {SPARROW = MAX + 1, PARROT = SPARROW + MAX};
4.    int main()
5.    {
6.        enum bird b = PARROT;
7.        printf("%d\n", b);
8.        return 0;
9.    }

```

- a) Compilation error
- b) 5
- c) Undefined value
- d) 2

View Answer

Answer: b

Explanation: MAX value is 2 and hence PARROT will have value 3 + 2.

Output:

```
$ cc pgm6.c
```

```
$ a.out
```

```
5
```

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  int main()
4.  {
5.      char *str = "x";
6.      char c = 'x';
7.      char ary[1];
8.      ary[0] = c;
9.      printf("%d %d", strlen(str), strlen(ary));
10.     return 0;
11. }
```

a) 1 1

b) 2 1

c) 2 2

d) 1 (undefined value)

View Answer

Answer: d

Explanation: str is null terminated, but ary is not null terminated.

Output:

```
$ cc pgm7.c
```

```
$ a.out
```

```
1 5
```

Constants – 2

1. enum types are processed by _____

a) Compiler

b) Preprocessor

c) Linker

d) Assembler

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        printf("sanfoundry\rclass\n");
5.        return 0;
6.    }
```

a) sanfoundryclass

b) sanfoundry

class

c) classundry

d) sanfoundry

[View Answer](#)

Answer: c

Explanation: r is carriage return and moves the cursor back. sanfo is replaced by class.

Output:

```
$ cc pgm8.c
```

```
$ a.out
```

```
classundry
```

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        printf("sanfoundry\r\nclass\n");
5.        return 0;
6.    }
```

a) sanfoundryclass

b) sanfoundry

class

c) classundry

d) sanfoundry

[View Answer](#)

Answer: b

Explanation: rn combination makes the cursor move to the next line.

Output:

```
$ cc pgm9.c
```

```
$ a.out
sanfoundry
class
```

4. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        const int p;
5.        p = 4;
6.        printf("p is %d", p);
7.        return 0;
8.    }
```

- a) p is 4
- b) Compile time error
- c) Run time error
- d) p is followed by a garbage value

View Answer

Answer: b

Explanation: Since the constant variable has to be declared and defined at the same time, not doing it results in an error.

Output:

```
$ cc pgm10.c
```

```
pgm10.c: In function 'main':
```

```
pgm10.c:5: error: assignment of read-only variable 'p'
```

5. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int k = 4;
5.        int *const p = &k;
6.        int r = 3;
7.        p = &r;
8.        printf("%d", p);
9.    }
```

- a) Address of k
- b) Address of r

- c) Compile time error
 - d) Address of k + address of r
- View Answer

Answer: c

Explanation: Since the pointer p is declared to be constant, trying to assign it with a new value results in an error.

Output:

```
$ cc pgm11.c
```

```
pgm11.c: In function 'main':
```

```
pgm11.c:7: error: assignment of read-only variable 'p'
```

```
pgm11.c:8: warning: format '%d' expects type 'int', but argument 2 has type 'int * const'
```

6. Which of the following statement is false?

- a) Constant variables need not be defined as they are declared and can be defined later
- b) Global constant variables are initialized to zero
- c) const keyword is used to define constant values
- d) You cannot reassign a value to a constant variable

View Answer

Answer: a

Explanation: Since the constant variable has to be declared and defined at the same time, not doing it results in an error.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int const k = 5;
5.      k++;
6.      printf("k is %d", k);
7.  }
```

- a) k is 6
- b) Error due to const succeeding int
- c) Error, because a constant variable can be changed only twice
- d) Error, because a constant variable cannot be changed

View Answer

Answer: d

Explanation: Constant variable has to be declared and defined at the same time. Trying to change it results in an error.

Output:

```
$ cc pgm12.c
```

```
pgm12.c: In function 'main':
```

```
pgm12.c:5: error: increment of read-only variable 'k'
```

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int const print()
3.  {
4.      printf("Sanfoundry.com");
5.      return 0;
6.  }
7.  void main()
8.  {
9.      print();
10. }
```

- a) Error because function name cannot be preceded by const
- b) Sanfoundry.com
- c) Sanfoundry.com is printed infinite times
- d) Blank screen, no output

[View Answer](#)

Answer: b

Explanation: None.

Output:

```
$ cc pgm13.c
```

```
$ a.out
```

```
Sanfoundry.com
```

Declarations – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo(const int *);
3.  int main()
4.  {
5.      const int i = 10;
6.      printf("%d ", i);
7.      foo(&i);
8.      printf("%d", i);
9.
10. }
```

```

11. void foo(const int *i)
12. {
13.     *i = 20;
14. }

```

- a) Compile time error
- b) 10 20
- c) Undefined value
- d) 10

View Answer

Answer: a

Explanation: Cannot change a const type value.

Output:

```
$ cc pgm1.c
```

```
pgm1.c: In function 'foo':
```

```
pgm1.c:13: error: assignment of read-only location '*i'
```

2. What will be the output of the following C code?

```

1. #include <stdio.h>
2. int main()
3. {
4.     const int i = 10;
5.     int *ptr = &i;
6.     *ptr = 20;
7.     printf("%d\n", i);
8.     return 0;
9. }

```

- a) Compile time error
- b) Compile time warning and printf displays 20
- c) Undefined behaviour
- d) 10

View Answer

Answer: b

Explanation: Changing const variable through non-constant pointers invokes compiler warning.

Output:

```
$ cc pgm2.c
```

```
pgm2.c: In function 'main':
```

```
pgm2.c:5: warning: initialization discards qualifiers from pointer target type
```

```
$ a.out
```

```
20
```

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      j = 10;
5.      printf("%d\n", j++);
6.      return 0;
7.  }
```

- a) 10
- b) 11
- c) Compile time error
- d) 0

View Answer

Answer: c

Explanation: Variable j is not defined.

Output:

```
$ cc pgm3.c
```

```
pgm3.c: In function 'main':
```

```
pgm3.c:4: error: 'j' undeclared (first use in this function)
```

```
pgm3.c:4: error: (Each undeclared identifier is reported only once
```

```
pgm3.c:4: error: for each function it appears in.)
```

4. Will the following C code compile without any error?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      for (int k = 0; k < 10; k++);
5.      return 0;
6.  }
```

- a) Yes
- b) No
- c) Depends on the C standard implemented by compilers
- d) Error

View Answer

Answer: c

Explanation: Compilers implementing C90 do not allow this, but compilers implementing C99 allow it.

Output:

```
$ cc pgm4.c
```

pgm4.c: In function 'main':

pgm4.c:4: error: 'for' loop initial declarations are only allowed in C99 mode

pgm4.c:4: note: use option -std=c99 or -std=gnu99 to compile your code

5. Will the following C code compile without any error?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int k;
5.      {
6.          int k;
7.          for (k = 0; k < 10; k++);
8.      }
9.  }
```

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the C standard implemented by compilers

View Answer

Answer: a

Explanation: There can be blocks inside the block. But within a block, variables have only block scope.

Output:

```
$ cc pgm5.c
```

6. Which of the following declaration is not supported by C?

- a) String str;
- b) char *str;
- c) float str = 3e2;
- d) Both "String str;" and "float str = 3e2;"

View Answer

Answer: a

Explanation: It is legal in Java, but not in C.

7. Which of the following format identifier can never be used for the variable var?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *var = "Advanced Training in C by Sanfoundry.com";
5.  }
```

- a) %f
- b) %d
- c) %c
- d) %s

View Answer

Answer: a

Explanation: %c can be used to print the indexed position.

%d can still be used to display its ASCII value.

%s is recommended.

%f cannot be used for the variable var.

Declarations – 2

1. Which of the following declaration is illegal?

- a) `char *str = "Best C programming classes by Sanfoundry";`
- b) `char str[] = "Best C programming classes by Sanfoundry";`
- c) `char str[20] = "Best C programming classes by Sanfoundry";`
- d) `char[] str = "Best C programming classes by Sanfoundry";`

View Answer

Answer: d

Explanation: `char[] str` is a declaration in Java, but not in C.

2. Which keyword is used to prevent any changes in the variable within a C program?

- a) `immutable`
- b) `mutable`
- c) `const`
- d) `volatile`

View Answer

Answer: c

Explanation: `const` is a keyword constant in C program.

3. Which of the following is not a pointer declaration?

- a) `char a[10];`
- b) `char a[] = {'1', '2', '3', '4'};`
- c) `char *str;`
- d) `char a;`

View Answer

Answer: d

Explanation: Array declarations are pointer declarations.

4. What will be the output of the following C code?

1. `#include <stdio.h>`
2. `void main()`
3. `{`
4. `int k = 4;`


```
5.     float k = 4;

6.     printf("%d", k)

7. }
```

- a) Compile time error
- b) 4
- c) 4.0000000
- d) 4.4

[View Answer](#)

Answer: a

Explanation: Since the variable k is defined both as integer and as float, it results in an error.

Output:

```
$ cc pgm8.c
```

```
pgm8.c: In function 'main':
```

```
pgm8.c:5: error: conflicting types for 'k'
```

```
pgm8.c:4: note: previous definition of 'k' was here
```

```
pgm8.c:6: warning: format '%d' expects type 'int', but argument 2 has type 'double'
```

```
pgm8.c:7: error: expected ';' before '}' token
```

5. Which of the following statement is false?

- a) A variable defined once can be defined again with different scope
- b) A single variable cannot be defined with two different types in the same scope
- c) A variable must be declared and defined at the same time
- d) A variable refers to a location in memory

[View Answer](#)

Answer: c

Explanation: It is not an error if the variable is declared and not defined. For example – extern declarations.

6. A variable declared in a function can be used in main().

- a) True
- b) False
- c) True if it is declared static
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: Since the scope of the variable declared within a function is restricted only within that function, so the above statement is false.

7. The name of the variable used in one function cannot be used in another function.

- a) True
- b) False

[View Answer](#)

Answer: b

Explanation: Since the scope of the variable declared within a function is restricted only within that function, the same name can be used to declare another variable in another function.

Arithmetic Operators – 1

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = -3;
5.        int k = i % 2;
6.        printf("%d\n", k);
7.    }
```

- a) Compile time error
- b) -1
- c) 1
- d) Implementation defined

[View Answer](#)

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 3;
5.        int l = i / -2;
6.        int k = i % -2;
7.        printf("%d %d\n", l, k);
8.        return 0;
9.    }
```

- a) Compile time error
- b) -1 1
- c) 1 -1
- d) Implementation defined

[View Answer](#)

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 5;
5.      i = i / 3;
6.      printf("%d\n", i);
7.      return 0;
8.  }
```

a) Implementation defined

b) 1

c) 3

d) Compile time error

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = -5;
5.      i = i / 3;
6.      printf("%d\n", i);
7.      return 0;
8.  }
```

a) Implementation defined

b) -1

c) -3

d) Compile time error

[View Answer](#)

Answer: b

Explanation: None.

5. What will be the final value of x in the following C code?

```
1.  #include <stdio.h>
2.  void main()
```

```
3.  {
4.      int x = 5 * 9 / 3 + 9;
5.  }
```

- a) 3.75
- b) Depends on compiler
- c) 24
- d) 3

[View Answer](#)

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  void main()
4.  {
5.      int x = 5.3 % 2;
6.      printf("Value of x is %d", x);
7.  }
```

- a) Value of x is 2.3
- b) Value of x is 1
- c) Value of x is 0.3
- d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  void main()
4.  {
5.      int y = 3;
6.      int x = 5 % 2 * 3 / 2;
7.      printf("Value of x is %d", x);
8.  }
```

- a) Value of x is 1
- b) Value of x is 2
- c) Value of x is 3

d) Compile time error

[View Answer](#)

Answer: a

Explanation: None.

Arithmetic Operators – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int a = 3;
5.        int b = ++a + a++ + --a;
6.        printf("Value of b is %d", b);
7.    }
```

a) Value of b is 12

b) Value of b is 13

c) Value of b is 10

d) Undefined behaviour

[View Answer](#)

Answer: d

Explanation: None.

2. What is the precedence of arithmetic operators (from highest to lowest)?

a) %, *, /, +, -

b) %, +, /, *, -

c) +, -, %, *, /

d) %, +, -, *, /

[View Answer](#)

Answer: a

Explanation: None.

3. Which of the following is not an arithmetic operation?

a) a * = 10;

b) a / = 10;

c) a ! = 10;

d) a % = 10;

[View Answer](#)

Answer: c

Explanation: None.

4. Which of the following data type will throw an error on modulus operation(%)?

a) char

b) short

c) int

d) float

[View Answer](#)

Answer: d

Explanation: None.

5. Which among the following are the fundamental arithmetic operators, i.e, performing the desired operation can be done using that operator only?

a) +, -

b) +, -, %

c) +, -, *, /

d) +, -, *, /, %

[View Answer](#)

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```
2.    int main()
```

```
3.    {
```

```
4.        int a = 10;
```

```
5.        double b = 5.6;
```

```
6.        int c;
```

```
7.        c = a + b;
```

```
8.        printf("%d", c);
```

```
9.    }
```

a) 15

b) 16

c) 15.6

d) 10

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```
2.    int main()
```

```
3.    {
```

```
4.        int a = 10, b = 5, c = 5;
```

```
5.     int d;  
6.     d = a == (b + c);  
7.     printf("%d", d);  
8.     }
```

a) Syntax error

b) 1

c) 10

d) 5

[View Answer](#)

Answer: b

Explanation: None.

Relational & Logical Operators – 1

1. What will be the output of the following C code?

```
1.     #include <stdio.h>  
2.     void main()  
3.     {  
4.         int x = 1, y = 0, z = 5;  
5.         int a = x && y || z++;  
6.         printf("%d", z);  
7.     }
```

a) 6

b) 5

c) 0

d) Varies

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.     #include <stdio.h>  
2.     void main()  
3.     {  
4.         int x = 1, y = 0, z = 5;  
5.         int a = x && y && z++;  
6.         printf("%d", z);
```

7. }

- a) 6
- b) 5
- c) 0
- d) Varies

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4.     int x = 1, y = 0, z = 3;
5.     x > y ? printf("%d", z) : return z;
6. }
```

- a) 3
- b) 1
- c) Compile time error
- d) Run time error

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1. #include <stdio.h>
2. void main()
3. {
4.     int x = 1, z = 3;
5.     int y = x << 3;
6.     printf(" %d\n", y);
7. }
```

- a) -2147483648
- b) -1
- c) Run time error
- d) 8

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 0, y = 2, z = 3;
5.      int a = x & y | z;
6.      printf("%d", a);
7.  }
```

- a) 3
- b) 0
- c) 2
- d) Run time error

View Answer

Answer: a

Explanation: None.

6. What will be the final value of j in the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 0;
5.      if (i && (j = i + 10))
6.          //do something
7.          ;
8.  }
```

- a) 0
- b) 10
- c) Depends on the compiler
- d) Depends on language standard

View Answer

Answer: a

Explanation: None.

7. What will be the final value of j in the following C code?

```
1.  #include <stdio.h>
```

```

2.   int main()
3.   {
4.       int i = 10, j = 0;
5.       if (i || (j = i + 10))
6.           //do something
7.       ;
8.   }

```

- a) 0
 - b) 20
 - c) Compile time error
 - d) Depends on language standard
- [View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int main()
3.   {
4.       int i = 1;
5.       if (i++ && (i == 1))
6.           printf("Yes\n");
7.       else
8.           printf("No\n");
9.   }

```

- a) Yes
 - b) No
 - c) Depends on the compiler
 - d) Depends on the standard
- [View Answer](#)

Answer: b

Explanation: None.

Relational & Logical Operators – 2

1. Are logical operator sequence points?

- a) True
- b) False

- c) Depends on the compiler
 - d) Depends on the standard
- View Answer

Answer: a

Explanation: None.

2. Do logical operators in the C language are evaluated with the short circuit?
- a) True
 - b) False
 - c) Depends on the compiler
 - d) Depends on the standard

View Answer

Answer: a

Explanation: None.

3. What is the result of logical or relational expression in C?
- a) True or False
 - b) 0 or 1
 - c) 0 if an expression is false and any positive number if an expression is true
 - d) None of the mentioned

View Answer

Answer: b

Explanation: None.

4. What will be the final value of d in the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int a = 10, b = 5, c = 5;
5.        int d;
6.        d = b + c == a;
7.        printf("%d", d);
8.    }
```

- a) Syntax error
- b) 1
- c) 5
- d) 10

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 10, b = 5, c = 3;
5.      b != !a;
6.      c = !!a;
7.      printf("%d\t%d", b, c);
8.  }

```

- a) 5 1
- b) 0 3
- c) 5 3
- d) 1 1

View Answer

Answer: a

Explanation: None.

6. Which among the following is NOT a logical or relational operator?

- a) !=
- b) ==
- c) ||
- d) =

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 10;
5.      if (a == a--)
6.          printf("TRUE 1\t");
7.      a = 10;
8.      if (a == --a)
9.          printf("TRUE 2\t");
10. }

```

- a) TRUE 1
- b) TRUE 2
- c) TRUE 1 TRUE 2
- d) Compiler Dependent

View Answer

Answer: d

Explanation: This is a sequence point problem and hence the result will be implementation dependent.

8. Relational operators cannot be used on _____

- a) structure
- b) long
- c) strings
- d) float

View Answer

Answer: a

Explanation: None.

Type Conversions – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      float x = 0.1;
5.      if (x == 0.1)
6.          printf("Sanfoundry");
7.      else
8.          printf("Advanced C Classes");
9.  }
```

- a) Advanced C Classes
- b) Sanfoundry
- c) Run time error
- d) Compile time error

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
```

```

3.  {
4.      float x = 0.1;
5.      printf("%d, ", x);
6.      printf("%f", x);
7.  }

```

- a) 0.100000, junk value
- b) Junk value, 0.100000
- c) 0, 0.100000
- d) 0, 0.999999

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code? (Initial values: x= 7, y = 8)

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      float x;
5.      int y;
6.      printf("enter two numbers \n");
7.      scanf("%f %f", &x, &y);
8.      printf("%f, %d", x, y);
9.  }

```

- a) 7.000000, 7
- b) Run time error
- c) 7.000000, junk
- d) Varies

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      double x = 123828749.66;

```

```
5.     int y = x;
6.     printf("%d\n", y);
7.     printf("%f\n", y);
8.     }
```

- a) 0, 0.0
- b) 123828749, 123828749.66
- c) 12382874, 12382874.0
- d) 123828749, 0.000000

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         int x = 97;
5.         char y = x;
6.         printf("%c\n", y);
7.     }
```

- a) a
- b) b
- c) 97
- d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

6. When double is converted to float, then the value is?

- a) Truncated
- b) Rounded
- c) Depends on the compiler
- d) Depends on the standard

[View Answer](#)

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
```

```
3.  {
4.      unsigned int i = 23;
5.      signed char c = -23;
6.      if (i > c)
7.          printf("Yes\n");
8.      else if (i < c)
9.          printf("No\n");
10. }
```

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the operating system

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 23;
5.      char c = -23;
6.      if (i < c)
7.          printf("Yes\n");
8.      else
9.          printf("No\n");
10. }
```

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

Type Conversions – 2

1. function `tolower(c)` defined in library `<ctype.h>` works for _____

- a) Ascii character set
- b) Unicode character set
- c) Ascii and utf-8 but not EBCDIC character set
- d) Any character set

[View Answer](#)

Answer: d

Explanation: None.

2. What will be the output of the following C code considering the size of a short int is 2, char is 1 and int is 4 bytes?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      short int i = 20;
5.      char c = 97;
6.      printf("%d, %d, %d\n", sizeof(i), sizeof(c), sizeof(c + i));
7.      return 0;
8.  }
```

- a) 2, 1, 2
- b) 2, 1, 1
- c) 2, 1, 4
- d) 2, 2, 8

[View Answer](#)

Answer: c

Explanation: None.

3. Which type of conversion is NOT accepted?

- a) From char to int
- b) From float to char pointer
- c) From negative int to char
- d) From double to char

[View Answer](#)

Answer: b

Explanation: Conversion of a float to pointer type is not allowed.

4. What will be the data type of the result of the following operation?

`(float)a * (int)b / (long)c * (double)d`

- a) int
- b) long

c) float
d) double
View Answer

Answer: d
Explanation: None.

5. Which of the following type-casting have chances for wrap around?
a) From int to float
b) From int to char
c) From char to short
d) From char to int
View Answer

Answer: b
Explanation: None.

6. Which of the following typecasting is accepted by C?
a) Widening conversions
b) Narrowing conversions
c) Widening & Narrowing conversions
d) None of the mentioned
View Answer

Answer: c
Explanation: None.

7. When do you need to use type-conversions?
a) The value to be stored is beyond the max limit
b) The value to be stored is in a form not supported by that data type
c) To reduce the memory in use, relevant to the value
d) All of the mentioned
View Answer

Answer: d
Explanation: None.

Increment and Decrement Operators – 1

1. What is the difference between the following 2 codes?

```
1.    #include <stdio.h> //Program 1
2.    int main()
3.    {
4.        int d, a = 1, b = 2;
5.        d = a++ + ++b;
6.        printf("%d %d %d", d, a, b);
7.    }
```

```

1.  #include <stdio.h> //Program 2
2.  int main()
3.  {
4.      int d, a = 1, b = 2;
5.      d = a++ ++b;
6.      printf("%d %d %d", d, a, b);
7.  }

```

- a) No difference as space doesn't make any difference, values of a, b, d are same in both the case
 - b) Space does make a difference, values of a, b, d are different
 - c) Program 1 has syntax error, program 2 is not
 - d) Program 2 has syntax error, program 1 is not
- View Answer

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 1, b = 1, c;
5.      c = a++ + b;
6.      printf("%d, %d", a, b);
7.  }

```

- a) a = 1, b = 1
- b) a = 2, b = 1
- c) a = 1, b = 2
- d) a = 2, b = 2

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 1, b = 1, d = 1;

```

5. `printf("%d, %d, %d", ++a + ++a+a++, a++ + ++b, ++d + d++ + a++);`

6. `}`

a) 15, 4, 5

b) 9, 6, 9

c) 9, 3, 5

d) Undefined (Compiler Dependent)

[View Answer](#)

Answer: d

Explanation: None.

4. For which of the following, "PI++;" code will fail?

a) `#define PI 3.14`

b) `char *PI = "A";`

c) `float PI = 3.14;`

d) none of the Mentioned

[View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `int main()`

3. `{`

4. `int a = 10, b = 10;`

5. `if (a = 5)`

6. `b--;`

7. `printf("%d, %d", a, b--);`

8. `}`

a) a = 10, b = 9

b) a = 10, b = 8

c) a = 5, b = 9

d) a = 5, b = 8

[View Answer](#)

Answer: c

Explanation: None.

6. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `int main()`

3. `{`

```
4.     int i = 0;
5.     int j = i++ + i;
6.     printf("%d\n", j);
7. }
```

- a) 0
- b) 1
- c) 2
- d) Compile time error

[View Answer](#)

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int i = 2;
5.         int j = ++i + i;
6.         printf("%d\n", j);
7.     }
```

- a) 6
- b) 5
- c) 4
- d) Compile time error

[View Answer](#)

Answer: a

Explanation: None.

8. Comment on the behaviour of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int i = 2;
5.         i = i++ + i;
6.         printf("%d\n", i);
7.     }
```

- a) = operator is not a sequence point
- b) ++ operator may return value with or without side effects
- c) it can be evaluated as (i++)+i or i+(++i)
- d) = operator is a sequence point

View Answer

Answer: a

Explanation: None.

Increment and Decrement Operators – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 0;
5.        int x = i++, y = ++i;
6.        printf("%d %d\n", x, y);
7.        return 0;
8.    }
```

- a) 0, 2
- b) 0, 1
- c) 1, 2
- d) Undefined

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 10;
5.        int *p = &i;
6.        printf("%d\n", *p++);
7.    }
```

- a) 10
- b) 11
- c) Garbage value

d) Address of i

[View Answer](#)

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int x = 97;
5.        int y = sizeof(x++);
6.        printf("X is %d", x);
7.    }
```

a) X is 97

b) X is 98

c) X is 99

d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int x = 4, y, z;
5.        y = --x;
6.        z = x--;
7.        printf("%d%d%d", x, y, z);
8.    }
```

a) 3 2 3

b) 2 3 3

c) 3 2 2

d) 2 3 4

[View Answer](#)

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 4;
5.      int *p = &x;
6.      int *k = p++;
7.      int r = p - k;
8.      printf("%d", r);
9.  }
```

- a) 4
- b) 8
- c) 1
- d) Run time error

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int a = 5, b = -7, c = 0, d;
5.      d = ++a && ++b || ++c;
6.      printf("\n%d%d%d%d", a, b, c, d);
7.  }
```

- a) 6 -6 0 0
- b) 6 -5 0 1
- c) -6 -6 0 1
- d) 6 -6 0 1

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
```



```
2. void main()
3. {
4.     int a = -5;
5.     int k = (a++, ++a);
6.     printf("%d\n", k);
7. }
```

- a) -4
- b) -5
- c) 4
- d) -3

View Answer

Answer: d

Explanation: None.

Bitwise Operators – 1

1. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4.     int c = 2 ^ 3;
5.     printf("%d\n", c);
6. }
```

- a) 1
- b) 8
- c) 9
- d) 0

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4.     unsigned int a = 10;
5.     a = ~a;
```

6. printf("%d\n", a);

7. }

a) -9

b) -10

c) -11

d) 10

[View Answer](#)

Answer: c

Explanation: None.

3. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. if (7 & 8)

5. printf("Honesty");

6. if ((~7 & 0x000f) == 8)

7. printf("is the best policy\n");

8. }

a) Honesty is the best policy

b) Honesty

c) is the best policy

d) No output

[View Answer](#)

Answer: c

Explanation: None.

4. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. int a = 2;

5. if (a >> 1)

6. printf("%d\n", a);

7. }

a) 0

b) 1

c) 2
d) No Output
View Answer

Answer: c
Explanation: None.

5. Comment on the output of the following C code.

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i, n, a = 4;
5.      scanf("%d", &n);
6.      for (i = 0; i < n; i++)
7.          a = a * 2;
8.  }
```

a) Logical Shift left
b) Logical Shift Right
c) Arithmetic Shift right
d) Bitwise exclusive OR
View Answer

Answer: a
Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 97;
5.      int y = sizeof(x++);
6.      printf("x is %d", x);
7.  }
```

a) x is 97
b) x is 98
c) x is 99
d) Run time error
View Answer

Answer: a
Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 4, y, z;
5.      y = --x;
6.      z = x--;
7.      printf("%d%d%d", x, y, z);
8.  }
```

- a) 3 2 3
- b) 2 2 3
- c) 3 2 2
- d) 2 3 3

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 4;
5.      int *p = &x;
6.      int *k = p++;
7.      int r = p - k;
8.      printf("%d", r);
9.  }
```

- a) 4
- b) 8
- c) 1
- d) Run time error

View Answer

Answer: c

Explanation: None.

Bitwise Operators – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int a = 5, b = -7, c = 0, d;
5.        d = ++a && ++b || ++c;
6.        printf("\n%d%d%d%d", a, b, c, d);
7.    }
```

- a) 6 -6 0 0
- b) 6 -5 0 1
- c) -6 -6 0 1
- d) 6 -6 0 1

[View Answer](#)

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int a = -5;
5.        int k = (a++, ++a);
6.        printf("%d\n", k);
7.    }
```

- a) -3
- b) -5
- c) 4
- d) Undefined

[View Answer](#)

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
```

```

3.  {
4.      int x = 2;
5.      x = x << 1;
6.      printf("%d\n", x);
7.  }

```

- a) 4
- b) 1
- c) Depends on the compiler
- d) Depends on the endianness of the machine

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = -2;
5.      x = x >> 1;
6.      printf("%d\n", x);
7.  }

```

- a) 1
- b) -1
- c) $2^{31} - 1$ considering int to be 4 bytes
- d) Either -1 or 1

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      if (~0 == 1)
5.          printf("yes\n");
6.      else

```

```
7.      printf("no\n");
8.      }
```

- a) yes
- b) no
- c) compile time error
- d) undefined

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.      #include <stdio.h>
2.      int main()
3.      {
4.          int x = -2;
5.          if (!0 == 1)
6.              printf("yes\n");
7.          else
8.              printf("no\n");
9.      }
```

- a) yes
- b) no
- c) run time error
- d) undefined

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.      #include <stdio.h>
2.      int main()
3.      {
4.          int y = 0;
5.          if (1 | (y = 1))
6.              printf("y is %d\n", y);
7.          else
```

8. `printf("%d\n", y);`

9.

10. `}`

a) y is 1

b) 1

c) run time error

d) undefined

[View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `int main()`

3. `{`

4. `int y = 1;`

5. `if (y & (y = 2))`

6. `printf("true %d\n", y);`

7. `else`

8. `printf("false %d\n", y);`

9.

10. `}`

a) true 2

b) false 2

c) either true 2 or false 2

d) true 1

[View Answer](#)

Answer: a

Explanation: None.

Assignment Operators & Expressions – 1

1. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `void main()`

3. `{`

4. `int x = 0;`


```

5.     if (x = 0)
6.         printf("Its zero\n");
7.     else
8.         printf("Its not zero\n");
9.     }

```

- a) Its not zero
- b) Its zero
- c) Run time error
- d) None

View Answer

Answer: a

Explanation: In the above C code, we assign a zero value to the variable x. In the if condition, we are reassigning a value of zero to x. Remember, we are “NOT” comparing its values to zero (you can see that it is a single ‘=’ sign, it’s not a double ‘==’ sign). Hence, the if-condition becomes false and the printf() function of the else condition will be executed which will display “Its not zero”.

2. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void main()
3.     {
4.         int k = 8;
5.         int x = 0 == 1 && k++;
6.         printf("%d%d\n", x, k);
7.     }

```

- a) 0 9
- b) 0 8
- c) 1 8
- d) 1 9

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void main()
3.     {
4.         char a = 'a';
5.         int x = (a % 10)++;

```

6. printf("%d\n", x);

7. }

a) 6

b) Junk value

c) Compile time error

d) 7

[View Answer](#)

Answer: c

Explanation: None.

4. What will be the output of the following C code snippet?

1. #include <stdio.h>

2. void main()

3. {

4. 1 < 2 ? return 1: return 2;

5. }

a) returns 1

b) returns 2

c) Varies

d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code snippet?

1. #include <stdio.h>

2. void main()

3. {

4. unsigned int x = -5;

5. printf("%d", x);

6. }

a) Run time error

b) Aries

c) -5

d) 5

[View Answer](#)

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 2, y = 1;
5.      x *= x + y;
6.      printf("%d\n", x);
7.      return 0;
8.  }
```

- a) 5
- b) 6
- c) Undefined behaviour
- d) Compile time error

[View Answer](#)

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 2, y = 2;
5.      x /= x / y;
6.      printf("%d\n", x);
7.      return 0;
8.  }
```

- a) 2
- b) 1
- c) 0.5
- d) Undefined behaviour

[View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```

2.    int main()
3.    {
4.        int x = 1, y = 0;
5.        x &&= y;
6.        printf("%d\n", x);
7.    }

```

- a) Compile time error
 - b) 1
 - c) 0
 - d) Undefined behaviour
- [View Answer](#)

Answer: a

Explanation: None.

Assignment Operators & Expressions – 2

1. What is the type of the following assignment expression if x is of type float and y is of type int?

`y = x + y;`

- a) int
- b) float
- c) there is no type for an assignment expression
- d) double

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the value of the following C expression?

`(x = foo()) != 1` considering `foo()` returns 2

- a) 2
- b) True
- c) 1
- d) 0

[View Answer](#)

Answer: c

Explanation: The C language sub-expression "`x = foo()`" will assign a value of 2 to the variable 'x'. Then, it will check if this value is not equal to 1 which is true and hence the result will be 1.

3. Operation "`a = a * b + a`" can also be written as _____

- a) `a *= b + 1;`
- b) `(c = a * b) != (a = c + a);`
- c) `a = (b + 1) * a;`

d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

4. What will be the final value of c in the following C statement? (Initial value: c = 2)

1. c <= 1;

a) c = 1;

b) c = 2;

c) c = 3;

d) c = 4;

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. int a = 1, b = 2;

5. a += b -= a;

6. printf("%d %d", a, b);

7. }

a) 1 1

b) 1 2

c) 2 1

d) 2 2

[View Answer](#)

Answer: c

Explanation: None.

6. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. int a = 4, n, i, result = 0;

5. scanf("%d", &n);

6. for (i = 0; i < n; i++)

7. result += a;

8. }

- a) Addition of a and n
- b) Subtraction of a and n
- c) Multiplication of a and n
- d) Division of a and n

View Answer

Answer: c

Explanation: None.

7. Which of the following is an invalid assignment operator?

- a) a %= 10;
- b) a /= 10;
- c) a |= 10;
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

Conditional Expressions – 1

1. What will be the output of the following C code?

```
1.        #include <stdio.h>

2.        int main()

3.        {

4.            int x = 2, y = 0;

5.            int z = (y++) ? y == 1 && x : 0;

6.            printf("%d\n", z);

7.            return 0;

8.        }
```

- a) 0
- b) 1
- c) Undefined behaviour
- d) Compile time error

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.        #include <stdio.h>
```

```

2.    int main()
3.    {
4.        int x = 1;
5.        int y = x == 1 ? getchar(): 2;
6.        printf("%d\n", y);
7.    }

```

- a) Compile time error
- b) Whatever character getchar function returns
- c) Ascii value of character getchar function returns
- d) 2

View Answer

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        int x = 1;
5.        short int i = 2;
6.        float f = 3;
7.        if (sizeof((x == 2) ? f : i) == sizeof(float))
8.            printf("float\n");
9.        else if (sizeof((x == 2) ? f : i) == sizeof(short int))
10.           printf("short int\n");
11.    }

```

- a) float
- b) short int
- c) Undefined behaviour
- d) Compile time error

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```

1.    #include <stdio.h>

```

```

2.   int main()
3.   {
4.       int a = 2;
5.       int b = 0;
6.       int y = (b == 0) ? a : (a > b) ? (b = 1): a;
7.       printf("%d\n", y);
8.   }

```

- a) Compile time error
 - b) 1
 - c) 2
 - d) Undefined behaviour
- [View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int main()
3.   {
4.       int y = 1, x = 0;
5.       int l = (y++, x++) ? y : x;
6.       printf("%d\n", l);
7.   }

```

- a) 1
 - b) 2
 - c) Compile time error
 - d) Undefined behaviour
- [View Answer](#)

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   void main()
3.   {
4.       int k = 8;

```



```

5.     int m = 7;

6.     int z = k < m ? k++ : m++;

7.     printf("%d", z);

8.     }

```

- a) 7
- b) 8
- c) Run time error
- d) 15

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

1.     #include <stdio.h>

2.     void main()

3.     {

4.         int k = 8;

5.         int m = 7;

6.         int z = k < m ? k = m : m++;

7.         printf("%d", z);

8.     }

```

- a) Run time error
- b) 7
- c) 8
- d) Depends on compiler

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```

1.     #include <stdio.h>

2.     void main()

3.     {

4.         1 < 2 ? return 1 : return 2;

5.     }

```

- a) returns 1
- b) returns 2

- c) Varies
 - d) Compile time error
- View Answer

Answer: d

Explanation: None.

Conditional Expressions – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int k = 8;
5.        int m = 7;
6.        k < m ? k++ : m = k;
7.        printf("%d", k);
8.    }
```

- a) 7
 - b) 8
 - c) Compile time error
 - d) Run time error
- View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int k = 8;
5.        int m = 7;
6.        k < m ? k = k + 1 : m = m + 1;
7.        printf("%d", k);
8.    }
```

- a) Compile time error
- b) 9
- c) 8

d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

3. What will be the final values of a and c in the following C statement? (Initial values: a = 2, c = 1)

```
c = (c) ? a = 0 : 2;
```

a) a = 0, c = 0;

b) a = 2, c = 2;

c) a = 2, c = 2;

d) a = 1, c = 2;

[View Answer](#)

Answer: a

Explanation: None.

4. What will be the data type of the following expression? (Initial data type: a = int, var1 = double, var2 = float)

```
expression (a < 50)? var1 : var2;
```

a) int

b) float

c) double

d) Cannot be determined

[View Answer](#)

Answer: c

Explanation: None.

5. Which expression has to be present in the following?

```
exp1 ? exp2 : exp3;
```

a) exp1

b) exp2

c) exp3

d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

6. What will be the final value of c in the following C code snippet? (Initial values: a = 1, b = 2, c = 1)

```
c += (-c) ? a : b;
```

a) Syntax Error

b) c = 1

c) c = 2

d) c = 3

[View Answer](#)

Answer: c

Explanation: None.

7. The following C code can be rewritten as _____

$c = (n) ? a : b;$

a)

if (!n)c = b;

else c = a;

b)

if (n <= 0)c = b;

else c = a;

c)

if (n > 0)c = a;

else c = b;

d) All of the mentioned

View Answer

Answer: a

Explanation: None.

Precedence and Order of Evaluation

1. Which of the following operators has an associativity from Right to Left?

a) <=

b) <<

c) ==

d) +=

View Answer

Answer: d

Explanation: None.

2. Which operators of the following have same precedence?

P. "!=" , Q. "+=" , R. "<="

a) P and Q

b) Q and R

c) P and R

d) P, Q and R

View Answer

Answer: b

Explanation: None.

3. Comment on the following statement.

```
n = 1;

printf("%d, %dn", 3*n, n++);
```

- a) Output will be 3, 2
- b) Output will be 3, 1
- c) Output will be 6, 1
- d) Output is compiler dependent

View Answer

Answer: d

Explanation: None.

4. Which of the following option is the correct representation of the following C statement?

```
e = a * b + c / d * f;
```

- a) `e = (a * (b +(c /(d * f))));`
- b) `e = ((a * b) + (c / (d * f)));`
- c) `e = ((a * b) + ((c / d)* f));`
- d) `e = (a * (b +((c / d) * f)));`

View Answer

Answer: c

Explanation: By using operator precedence and associativity rule. Also, verified by running `e = 5 * 4 + 6 / 3 * 2;` and printing the final value.

5. While swapping 2 numbers what precautions to be taken care?

```
b = (b / a);

a = a * b;

b = a / b;
```

- a) Data type should be either of short, int and long
- b) Data type should be either of float and double
- c) All data types are accepted except for (char *)
- d) This code doesn't swap 2 numbers

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include<stdio.h>

2.  int main()

3.  {

4.      int a = 1, b = 2, c = 3, d = 4, e;

5.      e = c + d = b * a;

6.      printf("%d, %d\n", e, d);
```

7. }

- a) 7, 4
- b) 7, 2
- c) 5, 2
- d) Syntax error

View Answer

Answer: d

Explanation: None.

7. Which of the following is the correct order of evaluation for the given expression?

$a = w \% x / y * z;$

- a) $\% / * =$
- b) $/ * \% =$
- c) $= \% * /$
- d) $* \% / =$

View Answer

Answer: a

Explanation: None.

8. Which function in the following expression will be called first?

$a = \text{func3}(6) - \text{func2}(4, 5) / \text{func1}(1, 2, 3);$

- a) $\text{func1}();$
- b) $\text{func2}();$
- c) $\text{func3}();$
- d) Cannot be predicted

View Answer

Answer: d

Explanation: None.

9. Which of the following operator has the highest precedence in the following?

- a) $()$
- b) sizeof
- c) $*$
- d) $+$

View Answer

Answer: a

Explanation: None.

10. Which of the following is a ternary operator?

- a) $\&\&$
- b) $>>=$
- c) $?:$
- d) $->$

View Answer

Answer: c

Explanation: None.

Precedence and Order of Evaluation – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int x = 2, y = 2;
5.        float f = y + x /= x / y;
6.        printf("%d %f\n", x, f);
7.        return 0;
8.    }
```

- a) 2 4.000000
- b) Compile time error
- c) 2 3.500000
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int x = 1, y = 2;
5.        if (x && y == 1)
6.            printf("true\n");
7.        else
8.            printf("false\n");
9.    }
```

- a) true
- b) false
- c) compile time error
- d) undefined behaviour

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 1, y = 2;
5.      int z = x & y == 2;
6.      printf("%d\n", z);
7.  }
```

a) 0

b) 1

c) Compile time error

d) Undefined behaviour

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 3, y = 2;
5.      int z = x /= y %= 2;
6.      printf("%d\n", z);
7.  }
```

a) 1

b) Compile time error

c) Floating point exception

d) Segmentation fault

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
```



```

3.  {
4.      int x = 3, y = 2;
5.      int z = x << 1 > 5;
6.      printf("%d\n", z);
7.  }

```

- a) 1
- b) 0
- c) 3
- d) Compile time error

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 3; //, y = 2;
5.      const int *p = &x;
6.      *p++;
7.      printf("%d\n", *p);
8.  }

```

- a) Increment of read-only location compile error
- b) 4
- c) Some garbage value
- d) Undefined behaviour

View Answer

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 2, y = 2;
5.      int z = x ^ y & 1;

```

6. printf("%d\n", z);

7. }

a) 1

b) 2

c) 0

d) 1 or 2

[View Answer](#)

Answer: b

Explanation: None.

8. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. int x = 2, y = 0;

5. int z = x && y = 1;

6. printf("%d\n", z);

7. }

a) 0

b) 1

c) Compile time error

d) 2

[View Answer](#)

Answer: c

Explanation: None.

9. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. int x = 0, y = 2;

5. if (!x && y)

6. printf("true\n");

7. else

8. printf("false\n");

9. }

- a) True
- b) False
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 0, y = 2;
5.      int z = ~x & y;
6.      printf("%d\n", z);
7.  }
```

- a) -1
- b) 2
- c) 0
- d) Compile time error

View Answer

Answer: b

Explanation: None.

Precedence and Order of Evaluation – 3

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int a = 5 * 3 + 2 - 4;
5.      printf("%d", a);
6.  }
```

- a) 13
- b) 14
- c) 12
- d) 16

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int a = 2 + 4 + 3 * 5 / 3 - 5;
5.        printf("%d", a);
6.    }
```

a) 7

b) 6

c) 10

d) 9

[View Answer](#)

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int a = 5 * 3 % 6 - 8 + 3;
5.        printf("%d", a);
6.    }
```

a) 10

b) 2

c) -2

d) -3

[View Answer](#)

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int b = 6;
```

```
5.     int c = 7;
6.     int a = ++b + c--;
7.     printf("%d", a);
8.     }
```

a) Run time error

b) 15

c) 13

d) 14

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main(
3.     {
4.         double b = 8;
5.         b++;
6.         printf("%lf", b);
7.     }
```

a) 9.000000

b) 9

c) 9.0

d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         double b = 3 % 0 * 1 - 4 / 2;
5.         printf("%lf", b);
6.     }
```

a) -2

b) Floating point Exception

c) 1

d) 0

[View Answer](#)

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      double b = 5 % 3 & 4 + 5 * 6;
5.      printf("%lf", b);
6.  }
```

a) 2

b) 30

c) 2.000000

d) Run time error

[View Answer](#)

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      double b = 3 && 5 & 4 % 3;
5.      printf("%lf", b);
6.  }
```

a) 3.000000

b) 4.000000

c) 5.000000

d) 1.000000

[View Answer](#)

Answer: d

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
```

```

3.  {
4.      double b = 5 & 3 && 4 || 5 | 6;
5.      printf("%lf", b);
6.  }

```

- a) 1.000000
- b) 0.000000
- c) 7.000000
- d) 2.000000

View Answer

Answer: a

Explanation: None.

10. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.
3.  void main()
4.  {
5.      int k = 0;
6.
7.      double b = k++ + ++k + k--;
8.
9.      printf("%d", k);
10. }

```

- a) 6
- b) 1
- c) 5
- d) undefined

View Answer

Answer: b

Explanation: None.

Precedence and Order of Evaluation – 4

1. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.
3.  void main()
4.  {
5.      int b = 5 - 4 + 2 * 5;
6.
7.      printf("%d", b);
8.  }

```

- a) 25
- b) -5
- c) 11
- d) 16

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int b = 5 & 4 & 6;
5.        printf("%d", b);
6.    }
```

- a) 5
- b) 6
- c) 3
- d) 4

View Answer

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int b = 5 & 4 | 6;
5.        printf("%d", b);
6.    }
```

- a) 6
- b) 4
- c) 1
- d) 0

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?


```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int b = 5 + 7 * 4 - 9 * (3, 2);
5.      printf("%d", b);
6.  }

```

- a) 6
- b) 15
- c) 13
- d) 21

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int h = 8;
5.      int b = (h++, h++);
6.      printf("%d%d\n", b, h);
7.  }

```

- a) 10 10
- b) 10 9
- c) 9 10
- d) 8 10

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int h = 8;
5.      int b = h++ + h++ + h++;

```

6. printf("%d\n", h);

7. }

a) 9

b) 10

c) 12

d) 11

[View Answer](#)

Answer: d

Explanation: None.

7. What will be the output of the following C code?

1. #include <stdio.h>

2. void main()

3. {

4. int h = 8;

5. int b = 4 * 6 + 3 * 4 < 3 ? 4 : 3;

6. printf("%d\n", b);

7. }

a) 3

b) 33

c) 34

d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code?

1. #include <stdio.h>

2. void main()

3. {

4. int a = 2 + 3 - 4 + 8 - 5 % 4;

5. printf("%d\n", a);

6. }

a) 0

b) 8

c) 11

d) 9

[View Answer](#)

Answer: b

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char a = '0';
5.      char b = 'm';
6.      int c = a && b || '1';
7.      printf("%d\n", c);
8.  }
```

- a) 0
- b) a
- c) 1
- d) m

[View Answer](#)

Answer: c

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char a = 'A';
5.      char b = 'B';
6.      int c = a + b % 3 - 3 * 2;
7.      printf("%d\n", c);
8.  }
```

- a) 65
- b) 58
- c) 64
- d) 59

[View Answer](#)

Answer: d

Explanation: None.

Precedence and Order of Evaluation – 5

1. What will be the output of the following C function?

```
1.  #include <stdio.h>
2.  void reverse(int i);
3.  int main()
4.  {
5.      reverse(1);
6.  }
7.  void reverse(int i)
8.  {
9.      if (i > 5)
10.         exit(0);
11.     printf("%d\n", i);
12.     return reverse(i++);
13. }
```

- a) 1 2 3 4 5
- b) 1 2 3 4
- c) Compile time error
- d) Stack overflow

View Answer

Answer: d

Explanation: None.

2. What will be the output of the following C function?

```
1.  #include <stdio.h>
2.  void reverse(int i);
3.  int main()
4.  {
5.      reverse(1);
6.  }
7.  void reverse(int i)
8.  {
9.      if (i > 5)
10.         return ;
```

```
11.    printf("%d ", i);
12.    return reverse((i++, i));
13. }
```

- a) 1 2 3 4 5
- b) Segmentation fault
- c) Compilation error
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

3. In expression $i = g() + f()$, first function called depends on _____

- a) Compiler
- b) Associativity of $()$ operator
- c) Precedence of $()$ and $+$ operator
- d) Left to right of the expression

View Answer

Answer: a

Explanation: None.

4. What will be the final values of i and j in the following C code?

```
1.    #include <stdio.h>
2.    int x = 0;
3.    int f()
4.    {
5.        if (x == 0)
6.            return ++x;
7.        else
8.            return --x;
9.    }
10.   int g()
11.   {
12.       return x++;
13.   }
14.   int main()
15.   {
16.       int i, j;
```

```

17.    i = (f() + g()) || g();
18.    x = 0;
19.    j = g() || (f() + g());
20.    }

```

- a) i value is 1 and j value is 1
- b) i value is 0 and j value is 0
- c) i value is 1 and j value is undefined
- d) i and j value are undefined

View Answer

Answer: d

Explanation: Logical OR are evaluated strictly from left-to-right. The '+' operator has two operands f() & g() and these functions f() and g() can get evaluated in any order depending on the compiler. When x is 0, (f() + g()) can return values of either 0 or 2 depending on the order of evaluation of these functions by the compiler. Hence, we can't have a fixed value for i and j.

5. What will be the final values of i and j in the following C code?

```

1.    #include <stdio.h>
2.
3.    int x = 0;
4.
5.    int f()
6.    {
7.        if (x == 0)
8.            return x + 1;
9.        else
10.           return x - 1;
11.    }
12.
13.    int g()
14.    {
15.        return x++;
16.    }
17.
18.    int main()
19.    {
20.        int i = (f() + g()) | g(); //bitwise or
21.        int j = g() | (f() + g()); //bitwise or
22.    }

```

- a) i value is 1 and j value is 1
- b) i value is 0 and j value is 0
- c) i value is 1 and j value is undefined
- d) i and j value are undefined

View Answer

Answer: d

Explanation: Bitwise OR are NOT strictly evaluated from left-to-right. Similar the '+' operator has two operands f() & g() and these functions f() and g() can get evaluated in any order depending on the compiler. Hence, we can't have a fixed value for i and j.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 2, y = 0;
5.      int z = y && (y |= 10);
6.      printf("%d\n", z);
7.      return 0;
8.  }
```

- a) 1
- b) 0
- c) Undefined behaviour due to order of evaluation
- d) 2

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 2, y = 0;
5.      int z = (y++) ? 2 : y == 1 && x;
6.      printf("%d\n", z);
7.      return 0;
8.  }
```

- a) 0
- b) 1

- c) 2
 - d) Undefined behaviour
- View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 2, y = 0;
5.      int z;
6.      z = (y++, y);
7.      printf("%d\n", z);
8.      return 0;
9.  }
```

- a) 0
 - b) 1
 - c) Undefined behaviour
 - d) Compilation error
- View Answer

Answer: b

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 2, y = 0, l;
5.      int z;
6.      z = y = 1, l = x && y;
7.      printf("%d\n", l);
8.      return 0;
9.  }
```

- a) 0
- b) 1

- c) Undefined behaviour due to order of evaluation can be different
- d) Compilation error

View Answer

Answer: b

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int y = 2;
5.      int z = y +(y = 10);
6.      printf("%d\n", z);
7.  }
```

- a) 12
- b) 20
- c) 4
- d) Either 12 or 20

View Answer

Answer: b

Explanation: None.

Precedence and Order of Evaluation – 6

1. Which of the following are unary operators?

- a) sizeof
- b) –
- c) ++
- d) all of the mentioned

View Answer

Answer: d

Explanation: None.

2. Where in C the order of precedence of operators do not exist?

- a) Within conditional statements, if, else
- b) Within while, do-while
- c) Within a macro definition
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

3. Associativity of an operator is _____

- a) Right to Left
- b) Left to Right
- c) Random fashion
- d) Both Right to Left and Left to Right

View Answer

Answer: d

Explanation: None.

4. Which of the following method is accepted for assignment?

- a) $5 = a = b = c = d;$
- b) $a = b = c = d = 5;$
- c) $a = b = 5 = c = d;$
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

5. Which of the following is NOT possible with any 2 operators in C?

- a) Different precedence, same associativity
- b) Different precedence, different associativity
- c) Same precedence, different associativity
- d) All of the mentioned

View Answer

Answer: c

Explanation: None.

6. Which of the following is possible with any 2 operators in C?

- a) Same associativity, different precedence
- b) Same associativity, same precedence
- c) Different associativity, different precedence
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

7. Which of the following operators has the lowest precedence?

- a) `!=`
- b) `&&`
- c) `?:`
- d) `,`

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```

2.    int main()
3.    {
4.        int x = 3, i = 0;
5.        do {
6.            x = x++;
7.            i++;
8.        } while (i != 3);
9.        printf("%d\n", x);
10.   }

```

a) Undefined behaviour

b) Output will be 3

c) Output will be 6

d) Output will be 5

View Answer

Answer: b

Explanation: None.

9. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        int a = -1, b = 4, c = 1, d;
5.        d = ++a && ++b || ++c;
6.        printf("%d, %d, %d, %d\n", a, b, c, d);
7.        return 0;
8.    }

```

a) 0, 4, 2, 1

b) 0, 5, 2, 1

c) -1, 4, 1, 1

d) 0, 5, 1, 0

View Answer

Answer: a

Explanation: None.

10. What will be the output of the following C code?

```

1.    #include <stdio.h>

```

```
2.   int main()
3.   {
4.       int p = 10, q = 20, r;
5.       if (r = p = 5 || q > 20)
6.           printf("%d", r);
7.       else
8.           printf("No Output\n");
9.   }
```

- a) 1
- b) 10
- c) 20
- d) No Output

[View Answer](#)

Answer: a

Explanation: None.

Multiple Choice Questions on Control Flow Statements in C

If-then-else Statements – 1

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int x = 5;
5.        if (x < 1)
6.            printf("hello");
7.        if (x == 5)
8.            printf("hi");
9.        else
10.           printf("no");
11.    }
```

- a) hi
 - b) hello
 - c) no
 - d) error
- [View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int x;
3.    void main()
4.    {
5.        if (x)
6.            printf("hi");
7.        else
8.            printf("how are u");
9.    }
```

- a) hi
- b) how are you

c) compile time error

d) error

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 5;
5.      if (true);
6.      printf("hello");
7.  }
```

a) It will display hello

b) It will throw an error

c) Nothing will be displayed

d) Compiler dependent

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 0;
5.      if (x == 0)
6.          printf("hi");
7.      else
8.          printf("how are u");
9.          printf("hello");
10. }
```

a) hi

b) how are you

c) hello

d) hihello

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 5;
5.      if (x < 1);
6.          printf("Hello");
7.
8.  }
```

a) Nothing

b) Run time error

c) Hello

d) Varies

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      double ch;
5.      printf("enter a value between 1 to 2:");
6.      scanf("%lf", &ch);
7.      switch (ch)
8.      {
9.          case 1:
10.             printf("1");
11.             break;
12.             case 2:
```

```
13.     printf("2");
14.     break;
15. }
16. }
```

- a) Compile time error
- b) 1
- c) 2
- d) Varies

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *ch;
5.      printf("enter a value between 1 to 3:");
6.      scanf("%s", ch);
7.      switch (ch)
8.      {
9.          case "1":
10.             printf("1");
11.             break;
12.          case "2":
13.             printf("2");
14.             break;
15.      }
16. }
```

- a) 1
- b) 2
- c) Compile time error
- d) No Compile time error

View Answer

Answer: c

Explanation: None.

8. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int ch;
5.      printf("enter a value between 1 to 2:");
6.      scanf("%d", &ch);
7.      switch (ch)
8.      {
9.          case 1:
10.             printf("1\n");
11.         default:
12.             printf("2\n");
13.     }
14. }
```

- a) 1
- b) 2
- c) 1 2
- d) Run time error

View Answer

Answer: c

Explanation: None.

9. What will be the output of the following C code? (Assuming that we have entered the value 2 in the standard input)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int ch;
5.      printf("enter a value between 1 to 2:");
6.      scanf("%d", &ch);
7.      switch (ch)
```

```

8.      {
9.      case 1:
10.      printf("1\n");
11.      break;
12.      printf("Hi");
13.      default:
14.      printf("2\n");
15.      }
16.  }

```

- a) 1
- b) Hi 2
- c) Run time error
- d) 2

View Answer

Answer: d

Explanation: None.

10. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int ch;
5.      printf("enter a value between 1 to 2:");
6.      scanf("%d", &ch);
7.      switch (ch, ch + 1)
8.      {
9.          case 1:
10.             printf("1\n");
11.             break;
12.          case 2:
13.             printf("2");
14.             break;
15.      }

```

16. }

- a) 1
- b) 2
- c) 3
- d) Run time error

View Answer

Answer: b

Explanation: None.

If-then-else Statements – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 1;
5.      if (x > 0)
6.          printf("inside if\n");
7.      else if (x > 0)
8.          printf("inside elseif\n");
9.  }
```

- a) inside if
- b) inside elseif
- c) inside if

inside elseif

- d) compile time error

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 0;
5.      if (x++)
6.          printf("true\n");
```

```
7.     else if (x == 1)
8.         printf("false\n");
9.     }
```

- a) true
- b) false
- c) compile time error
- d) undefined behaviour

[View Answer](#)

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int x = 0;
5.         if (x == 1)
6.             if (x == 0)
7.                 printf("inside if\n");
8.             else
9.                 printf("inside else if\n");
10.        else
11.            printf("inside else\n");
12.    }
```

- a) inside if
- b) inside else if
- c) inside else
- d) compile time error

[View Answer](#)

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
```

```

4.     int x = 0;
5.     if (x == 0)
6.         printf("true, ");
7.     else if (x = 10)
8.         printf("false, ");
9.     printf("%d\n", x);
10.    }

```

- a) false, 0
 - b) true, 0
 - c) true, 10
 - d) compile time error
- [View Answer](#)

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         int x = 0;
5.         if (x == 1)
6.             if (x >= 0)
7.                 printf("true\n");
8.         else
9.             printf("false\n");
10.    }

```

- a) true
 - b) false
 - c) Depends on the compiler
 - d) No print statement
- [View Answer](#)

Answer: d

Explanation: None.

6. The C statement `""if (a == 1 || b == 2) {}""` can be re-written as _____

a)

```
if (a == 1)
```

```
if (b == 2){}
```

b)

```
if (a == 1){}
```

```
if (b == 2){}
```

c)

```
if (a == 1){}
```

```
else if (b == 2){}
```

d) none of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

7. Which of the following is an invalid if-else statement?

a) if (if (a == 1)){}

b) if (func1 (a)){}

c) if (a){}

d) if ((char) a){}

[View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 1;
5.      if (a--)
6.          printf("True");
7.      if (a++)
8.          printf("False");
9.  }
```

a) True

b) False

c) True False

d) No Output

[View Answer](#)

Answer: a

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 1;
5.      if (a)
6.          printf("All is Well ");
7.          printf("I am Well\n");
8.      else
9.          printf("I am not a River\n");
10. }
```

- a) Output will be All is Well I am Well
 - b) Output will be I am Well I am not a River
 - c) Output will be I am Well
 - d) Compile time errors during compilation
- [View Answer](#)

Answer: d

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      if (printf("%d", printf(")))
5.          printf("We are Happy");
6.      else if (printf("1"))
7.          printf("We are Sad");
8.  }
```

- a) 0We are Happy
 - b) 1We are Happy
 - c) 1We are Sad
 - d) compile time error
- [View Answer](#)

Answer: d

Explanation: None.

Switch Statement

1. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      double ch;
5.      printf("enter a value between 1 to 2:");
6.      scanf("%lf", &ch);
7.      switch (ch)
8.      {
9.          case 1:
10.             printf("1");
11.             break;
12.          case 2:
13.             printf("2");
14.             break;
15.      }
16. }
```

a) Compile time error

b) 1

c) 2

d) Varies

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *ch;
5.      printf("enter a value between 1 to 3:");
```



```
6.    scanf("%s", ch);
7.    switch (ch)
8.    {
9.        case "1":
10.            printf("1");
11.            break;
12.        case "2":
13.            printf("2");
14.            break;
15.    }
16. }
```

- a) 1
 - b) Compile time error
 - c) 2
 - d) Run time error
- [View Answer](#)

Answer: b

Explanation: None.

3. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int ch;
5.        printf("enter a value between 1 to 2:");
6.        scanf("%d", &ch);
7.        switch (ch)
8.        {
9.            case 1:
10.                printf("1\n");
11.            default:
12.                printf("2\n");
13.        }
```

14. }

- a) 1
- b) 2
- c) 1 2
- d) Run time error

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code? (Assuming that we have entered the value 2 in the standard input)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int ch;
5.      printf("enter a value between 1 to 2:");
6.      scanf("%d", &ch);
7.      switch (ch)
8.      {
9.          case 1:
10.             printf("1\n");
11.             break;
12.             printf("hi");
13.         default:
14.             printf("2\n");
15.     }
16. }
```

- a) 1
- b) hi 2
- c) Run time error
- d) 2

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int ch;
5.      printf("enter a value between 1 to 2:");
6.      scanf("%d", &ch);
7.      switch (ch, ch + 1)
8.      {
9.          case 1:
10.             printf("1\n");
11.             break;
12.          case 2:
13.             printf("2");
14.             break;
15.      }
16.  }

```

- a) 1
- b) 2
- c) 3
- d) Run time error

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 1, b = 1;
5.      switch (a)
6.      {
7.          case a*b:
8.              printf("yes ");
9.          case a-b:

```

```
10.     printf("no\n");
11.     break;
12.     }
13.     }
```

- a) yes
- b) no
- c) Compile time error
- d) yes no

View Answer

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int x = 97;
5.         switch (x)
6.         {
7.             case 'a':
8.                 printf("yes ");
9.                 break;
10.            case 97:
11.                printf("no\n");
12.                break;
13.            }
14.        }
```

- a) yes
- b) yes no
- c) Duplicate case value error
- d) Character case value error

View Answer

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      float f = 1;
5.      switch (f)
6.      {
7.          case 1.0:
8.              printf("yes\n");
9.              break;
10.         default:
11.             printf("default\n");
12.     }
13. }
```

- a) yes
 - b) yes default
 - c) Undefined behaviour
 - d) Compile time error
- View Answer

Answer: d

Explanation: None.

Switch Statement - 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  const int a = 1, b = 2;
3.  int main()
4.  {
5.      int x = 1;
6.      switch (x)
7.      {
8.          case a:
9.              printf("yes ");
10.         case b:
11.             printf("no\n");
```

```
12.      break;
13.    }
14.  }
```

- a) yes no
 - b) yes
 - c) no
 - d) Compile time error
- [View Answer](#)

Answer: d

Explanation: We are violating a C programming rule which states that in switch-case statements, the labels must be integer constants. When you compile the code, the c compiler will give an error message indicating that the case label is not an integer constant because the labels given in the code are integer variables 'a' and 'b'. You will get the following error message:

error: case label does not reduce to an integer constant

case a:

error: case label does not reduce to an integer constant

case b:

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define max(a) a
3.  int main()
4.  {
5.      int x = 1;
6.      switch (x)
7.      {
8.          case max(2):
9.              printf("yes\n");
10.         case max(1):
11.             printf("no\n");
12.             break;
13.         }
14.     }
```

- a) yes no
- b) yes

- c) no
 - d) Compile time error
- View Answer

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      switch (printf("Do"))
5.      {
6.          case 1:
7.              printf("First\n");
8.              break;
9.          case 2:
10.             printf("Second\n");
11.             break;
12.          default:
13.              printf("Default\n");
14.              break;
15.      }
16. }
```

- a) Do
 - b) DoFirst
 - c) DoSecond
 - d) DoDefault
- View Answer

Answer: c

Explanation: None.

4. Comment on the output of the following C code.

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 1;
```

```

5.     switch (a)
6.     case 1:
7.         printf("%d", a);
8.     case 2:
9.         printf("%d", a);
10.    case 3:
11.        printf("%d", a);
12.    default:
13.        printf("%d", a);
14.    }

```

- a) No error, output is 1111
- b) No error, output is 1
- c) Compile time error, no break statements
- d) Compile time error, case label outside switch statement

View Answer

Answer: d

Explanation: None.

5. Which datatype can accept the switch statement?

- a) int
- b) char
- c) long
- d) all of the mentioned

View Answer

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        int a = 1;
5.        switch (a)
6.        {
7.            case a:
8.                printf("Case A ");
9.            default:

```



```
10.     printf("Default");
11.     }
12.     }
```

- a) Output: Case A
 - b) Output: Default
 - c) Output: Case A Default
 - d) Compile time error
- [View Answer](#)

Answer: d
Explanation: None.

7. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     switch (ch)
3.     {
4.         case 'a':
5.         case 'A':
6.             printf("true");
7.     }
```

- a) if (ch == 'a' && ch == 'A') printf("true");
- b)

if (ch == 'a')

if (ch == 'a') printf("true");

- c) if (ch == 'a' || ch == 'A') printf("true");
- d) none of the mentioned

[View Answer](#)

Answer: c
Explanation: None.

For Loops – 1

1. The C code 'for(;;)' represents an infinite loop. It can be terminated by _____

- a) break
- b) exit(0)
- c) abort()
- d) terminate

[View Answer](#)

Answer: a
Explanation: None.

2. What will be the correct syntax for running two variable for loop simultaneously?

a) for (i = 0; i < n; i++)

for (j = 0; j < n; j += 5)

b) for (i = 0, j = 0; i < n, j < n; i++, j += 5)

c) for (i = 0; i < n; i++){

for (j = 0; j < n; j += 5){}

d) none of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

3. Which for loop has range of similar indexes of 'i' used in for (i = 0; i < n; i++)?

a) for (i = n; i > 0; i--)

b) for (i = n; i >= 0; i--)

c) for (i = n-1; i > 0; i--)

d) for (i = n-1; i > -1; i--)

[View Answer](#)

Answer: d

Explanation: None.

4. Which of the following cannot be used as LHS of the expression in for (exp1; exp2; exp3)?

a) variable

b) function

c) typedef

d) macros

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      short i;
5.      for (i = 1; i >= 0; i++)
6.          printf("%d\n", i);
7.
8.  }
```

a) The control won't fall into the for loop

b) Numbers will be displayed until the signed limit of short and throw a runtime error

- c) Numbers will be displayed until the signed limit of short and program will successfully terminate
- d) This program will get into an infinite loop and keep printing numbers with no errors

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int k = 0;
5.      for (k)
6.          printf("Hello");
7.  }
```

- a) Compile time error
- b) hello
- c) Nothing
- d) Varies

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int k = 0;
5.      for (k < 3; k++)
6.          printf("Hello");
7.  }
```

- a) Compile time error
- b) Hello is printed thrice
- c) Nothing
- d) Varies

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      double k = 0;
5.      for (k = 0.0; k < 3.0; k++)
6.          printf("Hello");
7.  }

```

- a) Run time error
 - b) Hello is printed thrice
 - c) Hello is printed twice
 - d) Hello is printed infinitely
- [View Answer](#)

Answer: b

Explanation: None.

For Loops – 2

1. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      double k = 0;
5.      for (k = 0.0; k < 3.0; k++);
6.          printf("%lf", k);
7.  }

```

- a) 2.000000
 - b) 4.000000
 - c) 3.000000
 - d) Run time error
- [View Answer](#)

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {

```

```
4.     int k;
5.     for (k = -3; k < -5; k++)
6.         printf("Hello");
7.     }
```

- a) Hello
- b) Infinite hello
- c) Run time error
- d) Nothing

View Answer

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int i = 0;
5.         for ( ; ; )
6.             printf("In for loop\n");
7.             printf("After loop\n");
8.     }
```

- a) Compile time error
- b) Infinite loop
- c) After loop
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int i = 0;
5.         for (i++; i == 1; i = 2)
6.             printf("In for loop ");
```

7. `printf("After loop\n");`

8. `}`

- a) In for loop after loop
- b) After loop
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `int main()`

3. `{`

4. `int i = 0;`

5. `for (foo(); i == 1; i = 2)`

6. `printf("In for loop\n");`

7. `printf("After loop\n");`

8. `}`

9. `int foo()`

10. `{`

11. `return 1;`

12. `}`

- a) After loop
- b) In for loop after loop
- c) Compile time error
- d) Infinite loop

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `int main()`

3. `{`

4. `int *p = NULL;`

```

5.     for (foo(); p; p = 0)
6.         printf("In for loop\n");
7.         printf("After loop\n");
8.     }

```

- a) In for loop after loop
- b) Compile time error
- c) Infinite loop
- d) Depends on the value of NULL

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         for (int i = 0; i < 1; i++)
5.             printf("In for loop\n");
6.     }

```

- a) Compile time error
- b) In for loop
- c) Depends on the standard compiler implements
- d) Run time error

View Answer

Answer: c

Explanation: None.

While Loops – 1

1. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         while ()
5.             printf("In while loop ");
6.         printf("After loop\n");
7.     }

```

- a) In while loop after loop
- b) After loop
- c) Compile time error
- d) Infinite loop

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        do
5.            printf("In while loop ");
6.        while (0);
7.        printf("After loop\n");
8.    }
```

- a) In while loop
- b) In while loop After loop
- c) After loop
- d) Infinite loop

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 0;
5.        do {
6.            i++;
7.            printf("In while loop\n");
8.        } while (i < 3);
9.    }
```

- a)

In while loop

In while loop

In while loop

b)

In while loop

In while loop

c) Depends on the compiler

d) Compile time error

[View Answer](#)

Answer: a

Explanation: None.

4. How many times i value is checked in the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0;
5.      do {
6.          i++;
7.          printf("in while loop\n");
8.      } while (i < 3);
9.  }
```

a) 2

b) 3

c) 4

d) 1

[View Answer](#)

Answer: b

Explanation: None.

5. How many times i value is checked in the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0;
5.      while (i < 3)
```

```
6.      i++;  
7.      printf("In while loop\n");  
8.  }
```

- a) 2
- b) 3
- c) 4
- d) 1

[View Answer](#)

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>  
2.  void main()  
3.  {  
4.      int i = 2;  
5.      do  
6.      {  
7.          printf("Hi");  
8.      } while (i < 2)  
9.  }
```

- a) Compile time error
- b) Hi Hi
- c) Hi
- d) Varies

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>  
2.  void main()  
3.  {  
4.      int i = 0;  
5.      while (++i)  
6.      {
```

```
7.     printf("H");
8.     }
9.     }
```

- a) H
- b) H is printed infinite times
- c) Compile time error
- d) Varies

[View Answer](#)

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         int i = 0;
5.         do
6.         {
7.             printf("Hello");
8.         } while (i != 0);
9.     }
```

- a) Nothing
- b) H is printed infinite times
- c) Hello
- d) Run time error

[View Answer](#)

Answer: c

Explanation: None.

While Loops – 2

1. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         char *str = "";
5.         do
```

```
6.    {
7.        printf("hello");
8.    } while (str);
9.    }
```

- a) Nothing
- b) Run time error
- c) Varies
- d) Hello is printed infinite times

View Answer

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int i = 0;
5.      while (i < 10)
6.      {
7.          i++;
8.          printf("hi\n");
9.          while (i < 8)
10.         {
11.             i++;
12.             printf("hello\n");
13.         }
14.     }
15. }
```

- a) Hi is printed 8 times, hello 7 times and then hi 2 times
- b) Hi is printed 10 times, hello 7 times
- c) Hi is printed once, hello 7 times
- d) Hi is printed once, hello 7 times and then hi 2 times

View Answer

Answer: d

Explanation: None.

3. What is an example of iteration in C?

- a) for
- b) while
- c) do-while
- d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

4. How many times while loop condition is tested in the following C code snippets, if i is initialized to 0 in both the cases?

1. while (i < n)
2. i++;
3. -----
4. do
5. i++;
6. while (i <= n);

- a) n, n
- b) n, n+1
- c) n+1, n
- d) n+1, n+1

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code?

1. #include <stdio.h>
2. int main()
3. {
4. int i = 0;
5. while (i = 0)
6. printf("True\n");
7. printf("False\n");
8. }

- a) True (infinite time)
- b) True (1 time) False
- c) False
- d) Compiler dependent

[View Answer](#)

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 0;
5.      while (i < 5, j < 10)
6.      {
7.          i++;
8.          j++;
9.      }
10.     printf("%d, %d\n", i, j);
11. }
```

- a) 5, 5
- b) 5, 10
- c) 10, 10
- d) Syntax error

View Answer

Answer: c

Explanation: None.

7. Which loop is most suitable to first perform the operation and then test the condition?

- a) for loop
- b) while loop
- c) do-while loop
- d) none of the mentioned

View Answer

Answer: c

Explanation: None.

Break and Continue – 1

1. Which keyword can be used for coming out of recursion?

- a) break
- b) return
- c) exit
- d) both break and return

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 0, i = 0, b;
5.      for (i = 0; i < 5; i++)
6.      {
7.          a++;
8.          continue;
9.      }
10.     printf("%d", a);
11. }
```

a) 2

b) 3

c) 4

d) 5

[View Answer](#)

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 0, i = 0, b;
5.      for (i = 0; i < 5; i++)
6.      {
7.          a++;
8.          if (i == 3)
9.              break;
10.     }
11.     printf("%d", a);
```

12. }

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: None.

4. The keyword 'break' cannot be simply used within _____

- a) do-while
- b) if-else
- c) for
- d) while

View Answer

Answer: b

Explanation: None.

5. Which keyword is used to come out of a loop only for that iteration?

- a) break
- b) continue
- c) return
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int i = 0, j = 0;
5.      for (i = 0; i < 5; i++)
6.      {
7.          for (j = 0; j < 4; j++)
8.          {
9.              if (i > 1)
10.                 break;
11.          }
12.          printf("Hi \n");
```


13. }

14. }

- a) Hi is printed 5 times
- b) Hi is printed 9 times
- c) Hi is printed 7 times
- d) Hi is printed 4 times

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int i = 0;
5.      int j = 0;
6.      for (i = 0; i < 5; i++)
7.      {
8.          for (j = 0; j < 4; j++)
9.          {
10.             if (i > 1)
11.                 continue;
12.             printf("Hi \n");
13.         }
14.     }
15. }
```

- a) Hi is printed 9 times
- b) Hi is printed 8 times
- c) Hi is printed 7 times
- d) Hi is printed 6 times

[View Answer](#)

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```
2. void main()
3. {
4.     int i = 0;
5.     for (i = 0; i < 5; i++)
6.         if (i < 4)
7.             {
8.                 printf("Hello");
9.                 break;
10.            }
11. }
```

- a) Hello is printed 5 times
- b) Hello is printed 4 times
- c) Hello
- d) Hello is printed 3 times

View Answer

Answer: c

Explanation: None.

Break and Continue – 2

1. What will be the output of the following C code?

```
1. #include <stdio.h>
2. void main()
3. {
4.     int i = 0;
5.     if (i == 0)
6.         {
7.             printf("Hello");
8.             continue;
9.         }
10. }
```

- a) Hello is printed infinite times
- b) Hello
- c) Varies
- d) Compile time error

View Answer

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int i = 0;
5.      if (i == 0)
6.      {
7.          printf("Hello");
8.          break;
9.      }
10. }
```

a) Hello is printed infinite times

b) Hello

c) Varies

d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0;
5.      do
6.      {
7.          i++;
8.          if (i == 2)
9.              continue;
10.         printf("In while loop ");
11.     } while (i < 2);
12.     printf("%d\n", i);
```

13. }

- a) In while loop 2
- b) In while loop in while loop 3
- c) In while loop 3
- d) Infinite loop

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 0;
5.      for (i; i < 2; i++){
6.          for (j = 0; j < 3; j++)
7.              {
8.                  printf("1\n");
9.                  break;
10.             }
11.         printf("2\n");
12.     }
13.     printf("after loop\n");
14. }
```

a) 1

2

after loop

b) 1

after loop

c) 1

2

1

2

after loop

d) 1

1

2

after loop

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0;
5.      while (i < 2)
6.      {
7.          if (i == 1)
8.              break;
9.          i++;
10.         if (i == 1)
11.             continue;
12.         printf("In while loop\n");
13.     }
14.     printf("After loop\n");
15. }
```

a)

In while loop

After loop

b) After loop

c)

In while loop

In while loop

After loop

d) In while loop

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0;
5.      char c = 'a';
6.      while (i < 2)
7.      {
8.          i++;
9.          switch (c)
10.         {
11.             case 'a':
12.                 printf("%c ", c);
13.                 break;
14.                 break;
15.         }
16.     }
17.     printf("after loop\n");
18. }
```

a) a after loop

b) a a after loop

c) after loop

d) error

[View Answer](#)

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
```

```
3.  {
4.    printf("before continue ");
5.    continue;
6.    printf("after continue\n");
7. }
```

- a) Before continue after continue
- b) Before continue
- c) After continue
- d) Compile time error

View Answer

Answer: d

Explanation: None.

Goto & Labels – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.    printf("%d ", 1);
5.    goto l1;
6.    printf("%d ", 2);
7.    l1:goto l2;
8.    printf("%d ", 3);
9.    l2:printf("%d ", 4);
10. }
```

- a) 1 4
- b) Compilation error
- c) 1 2 4
- d) 1 3 4

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
```

```

3.  {
4.      printf("%d ", 1);
5.      l1:l2:
6.      printf("%d ", 2);
7.      printf("%d\n", 3);
8.  }

```

a) Compilation error

b) 1 2 3

c) 1 2

d) 1 3

[View Answer](#)

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      printf("%d ", 1);
5.      goto l1;
6.      printf("%d ", 2);
7.  }
8.  void foo()
9.  {
10.     l1 : printf("3 ", 3);
11. }

```

a) 1 2 3

b) 1 3

c) 1 3 2

d) Compilation error

[View Answer](#)

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>

```



```

2.   int main()
3.   {
4.       int i = 0, j = 0;
5.       while (i < 2)
6.       {
7.           l1 : i++;
8.           while (j < 3)
9.           {
10.              printf("Loop\n");
11.              goto l1;
12.          }
13.      }
14.  }

```

- a) Loop Loop
 - b) Compilation error
 - c) Loop Loop Loop Loop
 - d) Infinite Loop
- View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int main()
3.   {
4.       int i = 0, j = 0;
5.       while (l1: i < 2)
6.       {
7.           i++;
8.           while (j < 3)
9.           {
10.              printf("loop\n");
11.              goto l1;
12.          }

```

13. }

14. }

- a) loop loop
- b) Compilation error
- c) loop loop loop loop
- d) Infinite loop

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 0;
5.      l1: while (i < 2)
6.      {
7.          i++;
8.          while (j < 3)
9.          {
10.             printf("loop\n");
11.             goto l1;
12.          }
13.      }
14. }
```

- a) loop loop
- b) compilation error
- c) oop loop loop loop
- d) infinite loop

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
```

```

3.  {
4.      int i = 0;
5.      if (i == 0)
6.      {
7.          goto label;
8.      }
9.      label: printf("Hello");
10. }

```

- a) Nothing
- b) Error
- c) Infinite Hello
- d) Hello

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int i = 0, k;
5.      if (i == 0)
6.          goto label;
7.      for (k = 0; k < 3; k++)
8.      {
9.          printf("hi ");
10.         label: k = printf("%03d", i);
11.     }
12. }

```

- a) 0
- b) hi hi hi 000
- c) 0 hi hi hi 000
- d) 000

View Answer

Answer: d

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int i = 0, k;
5.      label: printf("%d", i);
6.      if (i == 0)
7.          goto label;
8.  }
```

- a) 0
 - b) Infinite 0
 - c) Nothing
 - d) Error
- View Answer

Answer: b

Explanation: None

Goto & Labels – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int i = 5, k;
5.      if (i == 0)
6.          goto label;
7.      label: printf("%d", i);
8.      printf("Hey");
9.  }
```

- a) 5
 - b) Hey
 - c) 5 Hey
 - d) Nothing
- View Answer

Answer: c

Explanation: None.

2. goto can be used to jump from main() to within a function.

- a) true
- b) false
- c) depends
- d) varies

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      printf("%d ", 1);
5.      goto l1;
6.      printf("%d ", 2);
7.      l1:goto l2;
8.      printf("%d ", 3);
9.      l2:printf("%d ", 4);
10. }
```

- a) 1 4
- b) Compile time error
- c) 1 2 4
- d) 1 3 4

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      printf("%d ", 1);
5.      l1:l2:
6.      printf("%d ", 2);
```

7. printf("%d\n", 3);

8. }

a) Compile time error

b) 1 2 3

c) 1 2

d) 1 3

[View Answer](#)

Answer: b

Explanation: None.

5. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. printf("%d ", 1);

5. goto l1;

6. printf("%d ", 2);

7. }

8. void foo()

9. {

10. l1: printf("3 ", 3);

11. }

a) 1 2 3

b) 1 3

c) 1 3 2

d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

6. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. int i = 0, j = 0;

5. while (i < 2)

```

6.      {
7.      l1: i++;
8.      while (j < 3)
9.      {
10.         printf("loop\n");
11.         goto l1;
12.     }
13. }
14. }

```

- a) loop loop
- b) Compile time error
- c) loop loop loop loop
- d) Infinite loop

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 0;
5.      while (l1: i < 2)
6.      {
7.          i++;
8.          while (j < 3)
9.          {
10.             printf("loop\n");
11.             goto l1;
12.         }
13.     }
14. }

```

- a) loop loop
- b) Compile time error

c) loop loop loop loop

d) Infinite loop

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 0;
5.      l1: while (i < 2)
6.          {
7.              i++;
8.              while (j < 3)
9.                  {
10.                     printf("loop\n");
11.                     goto l1;
12.                  }
13.          }
14. }
```

a) loop loop

b) Compile time error

c) loop loop loop loop

d) Infinite loop

View Answer

Answer: a

Explanation: None.

MCQs on C Functions and Structure of a Program

Basics of Functions – 1

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        m();
5.        void m()
6.        {
7.            printf("hi");
8.        }
9.    }
```

- a) hi
- b) Compile time error
- c) Nothing
- d) Varies

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        m();
5.    }
6.    void m()
7.    {
8.        printf("hi");
9.        m();
10.   }
```

- a) Compile time error
- b) hi
- c) Infinite hi

d) Nothing

[View Answer](#)

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      static int x = 3;
5.      x++;
6.      if (x <= 5)
7.      {
8.          printf("hi");
9.          main();
10.     }
11. }
```

a) Run time error

b) hi

c) Infinite hi

d) hi hi

[View Answer](#)

Answer: d

Explanation: None.

4. Which of the following is a correct format for declaration of function?

a) return-type function-name(argument type);

b) return-type function-name(argument type){}

c) return-type (argument type)function-name;

d) all of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

5. Which of the following function declaration is illegal?

a) int 1bhk(int);

b) int 1bhk(int a);

c) int 2bhk(int*, int []);

d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

6. Which function definition will run correctly?

a)

```
int sum(int a, int b)

return (a + b);
```

b)

```
int sum(int a, int b)

{return (a + b);}
```

c)

```
int sum(a, b)

return (a + b);
```

d) none of the mentioned

View Answer

Answer: b

Explanation: None.

7. Can we use a function as a parameter of another function? [Eg: void wow(int func())].

a) Yes, and we can use the function value conveniently

b) Yes, but we call the function again to get the value, not as convenient as in using variable

c) No, C does not support it

d) This case is compiler dependent

View Answer

Answer: c

Explanation: None.

8. The value obtained in the function is given back to main by using _____ keyword.

a) return

b) static

c) new

d) volatile

View Answer

Answer: a

Explanation: None.

Basics of Functions – 2

1. What will be the output of the following C code?

1. `#include <stdio.h>`
2. `int main()`

```

3.  {
4.      void foo();
5.      printf("1 ");
6.      foo();
7.  }
8.  void foo()
9.  {
10.     printf("2 ");
11. }

```

- a) 1 2
- b) Compile time error
- c) 1 2 1 2
- d) Depends on the compiler

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      void foo(), f();
5.      f();
6.  }
7.  void foo()
8.  {
9.      printf("2 ");
10. }
11. void f()
12. {
13.     printf("1 ");
14.     foo();
15. }

```

- a) Compile time error as foo is local to main
- b) 1 2
- c) 2 1
- d) Compile time error due to declaration of functions inside main

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      void foo();
5.      void f()
6.      {
7.          foo();
8.      }
9.      f();
10. }
11. void foo()
12. {
13.     printf("2 ");
14. }
```

- a) 2 2
- b) 2
- c) Compile time error
- d) Depends on the compiler

View Answer

Answer: d

Explanation: Even though the answer is 2, this code will compile fine only with gcc. GNU C supports nesting of functions in C as a language extension whereas standard C compiler doesn't.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo();
3.  int main()
4.  {
```

```
5.    void foo();
6.    foo();
7.    return 0;
8.    }
9.    void foo()
10.   {
11.       printf("2 ");
12.   }
```

- a) Compile time error
 - b) 2
 - c) Depends on the compiler
 - d) Depends on the standard
- [View Answer](#)

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void foo();
3.    int main()
4.    {
5.        void foo(int);
6.        foo(1);
7.        return 0;
8.    }
9.    void foo(int i)
10.   {
11.       printf("2 ");
12.   }
```

- a) 2
 - b) Compile time error
 - c) Depends on the compiler
 - d) Depends on the standard
- [View Answer](#)

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo();
3.  int main()
4.  {
5.      void foo(int);
6.      foo();
7.      return 0;
8.  }
9.  void foo()
10. {
11.     printf("2 ");
12. }
```

a) 2

b) Compile time error

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void m()
3.  {
4.      printf("hi");
5.  }
6.  void main()
7.  {
8.      m();
9.  }
```

- a) hi
- b) Run time error
- c) Nothing
- d) Varies

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void m();
3.  void n()
4.  {
5.      m();
6.  }
7.  void main()
8.  {
9.      void m()
10.     {
11.         printf("hi");
12.     }
13. }
```

- a) hi
- b) Compile time error
- c) Nothing
- d) Varies

View Answer

Answer: b

Explanation: None.

Functions Returning Non-integers – 1

1. What is the return-type of the function sqrt()?
- a) int
 - b) float
 - c) double
 - d) depends on the data type of the parameter

View Answer

Answer: c

Explanation: None.

2. Which of the following function declaration is illegal?

a)

```
double func();
```

```
int main(){}  
double func(){}  
b)
```

```
double func(){};
```

```
int main(){}  
c)
```

```
int main()  
{  
    double func();  
}
```

```
double func(){}  
d) None of the mentioned
```

```
double func(){}  
View Answer
```

Answer: d

Explanation: None.

3. What will be the output of the following C code having void return-type function?

```
1. #include <stdio.h>
```

```
2. void foo()
```

```
3. {
```

```
4.     return 1;
```

```
5. }
```

```
6. void main()
```

```
7. {
```

```
8.     int x = 0;
```

```
9.     x = foo();
```

```
10.    printf("%d", x);
```

```
11. }
```

- a) 1
- b) 0
- c) Runtime error
- d) Compile time error

View Answer

Answer: d

Explanation: None.

4. What will be the data type returned for the following C function?

```
1.    #include <stdio.h>
2.    int func()
3.    {
4.        return (double)(char)5.0;
5.    }
```

- a) char
- b) int
- c) double
- d) multiple type-casting in return is illegal

View Answer

Answer: b

Explanation: None.

5. What is the problem in the following C declarations?

```
int func(int);

double func(int);

int func(float);
```

- a) A function with same name cannot have different signatures
- b) A function with same name cannot have different return types
- c) A function with same name cannot have different number of parameters
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int m()
3.    {
4.        printf("hello");
```

```
5.    }  
6.    void main()  
7.    {  
8.        int k = m();  
9.        printf("%d", k);  
10.   }
```

- a) hello5
 - b) Error
 - c) Nothing
 - d) Junk value
- View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.    #include <stdio.h>  
2.    int *m()  
3.    {  
4.        int *p = 5;  
5.        return p;  
6.    }  
7.    void main()  
8.    {  
9.        int *k = m();  
10.    printf("%d", k);  
11.   }
```

- a) 5
 - b) Junk value
 - c) 0
 - d) Error
- View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```

2.   int *m();
3.   void main()
4.   {
5.       int *k = m();
6.       printf("hello ");
7.       printf("%d", k[0]);
8.   }
9.   int *m()
10.  {
11.      int a[2] = {5, 8};
12.      return a;
13.  }

```

- a) hello 5 8
- b) hello 5
- c) hello followed by garbage value or runtime error
- d) Compilation error

View Answer

Answer: c

Explanation: Since are returning the address of a local array variable 'a', the compiler will give a warning in most of the cases. However, the program will run, but it will result in unexpected behaviour. In most of the compilers, it will result in a core dump due to segmentation fault whereas in some compilers, it might print hello followed by garbage value.

Functions Returning Non-integers – 2

1. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int *m();
3.   void main()
4.   {
5.       int k = m();
6.       printf("%d", k);
7.   }
8.   int *m()
9.   {
10.      int a[2] = {5, 8};

```

11. return a;

12. }

a) 5

b) 8

c) Nothing

d) Varies

[View Answer](#)

Answer: d

Explanation: None.

2. What will be the output of the following C code?

1. #include <stdio.h>

2. void m(int k)

3. {

4. printf("hi");

5. }

6. void m(double k)

7. {

8. printf("hello");

9. }

10. void main()

11. {

12. m(3);

13. }

a) hi

b) hello

c) Compile time error

d) Nothing

[View Answer](#)

Answer: c

Explanation: None.

3. What is the default return type if it is not specified in function definition?

a) void

b) int

c) double

d) short int

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int foo();
3.  int main()
4.  {
5.      int i = foo();
6.  }
7.  foo()
8.  {
9.      printf("2 ");
10.     return 2;
11. }
```

a) 2

b) Compile time error

c) Depends on the compiler

d) Depends on the standard

[View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  double foo();
3.  int main()
4.  {
5.      foo();
6.      return 0;
7.  }
8.  foo()
9.  {
10.     printf("2 ");
11.     return 2;
```

12. }

- a) 2
- b) Compile time error
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

6. Functions can return structure in C?

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

7. Functions can return enumeration constants in C?

- a) true
- b) false
- c) depends on the compiler
- d) depends on the standard

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  enum m{JAN, FEB, MAR};
3.  enum m foo();
4.  int main()
5.  {
6.      enum m i = foo();
7.      printf("%d\n", i);
8.  }
9.  int foo()
10. {
11.     return JAN;
12. }
```

- a) Compile time error
- b) 0
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

External Variables – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      m();
5.      printf("%d", x);
6.  }
7.  int x;
8.  void m()
9.  {
10.     x = 4;
11. }
```

- a) 4
- b) Compile time error
- c) 0
- d) Undefined

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int x;
3.  void main()
4.  {
5.      printf("%d", x);
6.  }
```


- a) Junk value
- b) Run time error
- c) 0
- d) Undefined

View Answer

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int x = 5;
3.  void main()
4.  {
5.      int x = 3;
6.      printf("%d", x);
7.      {
8.          x = 4;
9.      }
10.     printf("%d", x);
11. }
```

- a) Run time error
- b) 3 3
- c) 3 5
- d) 3 4

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int x = 5;
3.  void main()
4.  {
5.      int x = 3;
6.      printf("%d", x);
7.      {
```

```
8.      int x = 4;
9.      }
10.     printf("%d", x);
11.     }
```

- a) 3 3
- b) 3 4
- c) 3 5
- d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

5. Functions in C are always _____

- a) Internal
- b) External
- c) Both Internal and External
- d) External and Internal are not valid terms for functions

[View Answer](#)

Answer: b

Explanation: None.

6. Global variables are _____

- a) Internal
- b) External
- c) Both Internal and External
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

7. Which of the following is an external variable in the following C code?

```
1.  #include <stdio.h>
2.  int func (int a)
3.  {
4.      int b;
5.      return b;
6.  }
7.  int main()
8.  {
9.      int c;
```

10. func (c);

11. }

12. int d;

a) a

b) b

c) c

d) d

[View Answer](#)

Answer: d

Explanation: None.

8. What will be the output of the following C code?

1. #include <stdio.h>

2. int main()

3. {

4. printf("%d", d++);

5. }

6. int d = 10;

a) 9

b) 10

c) 11

d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

9. What will be the output of the following C code?

1. #include <stdio.h>

2. double var = 8;

3. int main()

4. {

5. int var = 5;

6. printf("%d", var);

7. }

a) 5

b) 8

c) Compile time error due to wrong format identifier for double

d) Compile time error due to redeclaration of variable with same name

[View Answer](#)

Answer: a

Explanation: None.

External Variables – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    double i;
3.    int main()
4.    {
5.        printf("%g\n",i);
6.        return 0;
7.    }
```

a) 0

b) 0.000000

c) Garbage value

d) Depends on the compiler

[View Answer](#)

Answer: a

Explanation: None.

2. Which part of the program address space is p stored in the following C code?

```
1.    #include <stdio.h>
2.    int *p = NULL;
3.    int main()
4.    {
5.        int i = 0;
6.        p = &i;
7.        return 0;
8.    }
```

a) Code/text segment

b) Data segment

c) Bss segment

d) Stack

[View Answer](#)

Answer: b

Explanation: None.

3. Which part of the program address space is p stored in the following C code?

```
1.  #include <stdio.h>
2.  int *p;
3.  int main()
4.  {
5.      int i = 0;
6.      p = &i;
7.      return 0;
8.  }
```

- a) Code/text segment
- b) Data segment
- c) Bss segment
- d) Stack

View Answer

Answer: c

Explanation: None.

4. Can variable i be accessed by functions in another source file?

```
1.  #include <stdio.h>
2.  int i;
3.  int main()
4.  {
5.      printf("%d\n", i);
6.  }
```

- a) Yes
- b) No
- c) Only if static keyword is used
- d) Depends on the type of the variable

View Answer

Answer: a

Explanation: None.

5. Property of the external variable to be accessed by any source file is called by the C90 standard as

-
- a) external linkage
 - b) external scope

- c) global scope
 - d) global linkage
- View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int *i;
3.  int main()
4.  {
5.      if (i == NULL)
6.          printf("true\n");
7.      return 0;
8.  }
```

- a) true
- b) true only if NULL value is 0
- c) Compile time error
- d) Nothing

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int *i;
3.  int main()
4.  {
5.      if (i == 0)
6.          printf("true\n");
7.      return 0;
8.  }
```

- a) true
- b) true only if NULL value is 0
- c) Compile time error
- d) Nothing

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  static int x = 5;
3.  void main()
4.  {
5.      x = 9;
6.      {
7.          int x = 4;
8.      }
9.      printf("%d", x);
10. }
```

a) 9

b) 4

c) 5

d) 0

[View Answer](#)

Answer: a

Explanation: None.

Scope of a Variable – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int i;
3.  int main()
4.  {
5.      extern int i;
6.      if (i == 0)
7.          printf("scope rules\n");
8.  }
```

a) scope rules

b) Compile time error due to multiple declaration

c) Compile time error due to not defining type in statement extern i

d) Nothing will be printed as value of i is not zero because i is an automatic variable

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code (without linking the source file in which ary1 is defined)?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      extern ary1[];
5.      printf("scope rules\n");
6.  }
```

a) scope rules

b) Linking error due to undefined reference

c) Compile time error because size of array is not provided

d) Compile time error because datatype of array is not provided

[View Answer](#)

Answer: a

Explanation: None.

3. What will be the output of the following C code (after linking to source file having definition of ary1)?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      extern ary1[];
5.      printf("%d\n", ary1[0]);
6.  }
```

a) Value of ary1[0];

b) Compile time error due to multiple definition

c) Compile time error because size of array is not provided

d) Compile time error because datatype of array is not provided

[View Answer](#)

Answer: d

Explanation: None.

4. What is the scope of an external variable?

a) Whole source file in which it is defined

b) From the point of declaration to the end of the file in which it is defined

- c) Any source file in a program
 - d) From the point of declaration to the end of the file being compiled
- View Answer

Answer: d

Explanation: None.

5. What is the scope of a function?

- a) Whole source file in which it is defined
- b) From the point of declaration to the end of the file in which it is defined
- c) Any source file in a program
- d) From the point of declaration to the end of the file being compiled

View Answer

Answer: d

Explanation: None.

6. Comment on the output of the following C code.

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i;
5.      for (i = 0; i < 5; i++)
6.          int a = i;
7.      printf("%d", a);
8.  }
```

- a) a is out of scope when printf is called
- b) Redclaration of a in same scope throws error
- c) Syntax error in declaration of a
- d) No errors, program will show the output 5

View Answer

Answer: c

Explanation: None.

7. Which variable has the longest scope in the following C code?

```
1.  #include <stdio.h>
2.  int b;
3.  int main()
4.  {
5.      int c;
6.      return 0;
```

```
7.    }  
8.    int a;
```

- a) a
- b) b
- c) c
- d) Both a and b

View Answer

Answer: b

Explanation: None.

8. Comment on the following 2 C programs.

```
1.    #include <stdio.h> //Program 1  
2.    int main()  
3.    {  
4.        int a;  
5.        int b;  
6.        int c;  
7.    }
```

8.

```
9.    #include <stdio.h> //Program 2  
10.   int main()  
11.   {  
12.       int a;  
13.       {  
14.           int b;  
15.       }  
16.       {  
17.           int c;  
18.       }  
19.   }
```

- a) Both are same
- b) Scope of c is till the end of the main function in Program 2
- c) In Program 1, variables a, b and c can be used anywhere in the main function whereas in Program 2, variables b and c can be used only inside their respective blocks

d) None of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

Scope of a Variable – 2

1. What will be the sequence of allocation and deletion of variables in the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int a;
5.        {
6.            int b;
7.        }
8.    }
```

a) a->b, a->b

b) a->b, b->a

c) b->a, a->b

d) b->a, b->a

[View Answer](#)

Answer: b

Explanation: None.

2. Array sizes are optional during array declaration by using _____ keyword.

a) auto

b) static

c) extern

d) register

[View Answer](#)

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int x = 3;
5.        {
```

```
6.      x = 4;
7.      printf("%d", x);
8.  }
```

9. }
- a) 4
b) 3
c) 0
d) Undefined

[View Answer](#)

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int x = 5;
3.  void main()
4.  {
5.      int x = 3;
6.      m();
7.      printf("%d", x);
8.  }
9.  void m()
10. {
11.     x = 8;
12.     n();
13. }
14. void n()
15. {
16.     printf("%d", x);
17. }
```

- a) 8 3
b) 3 8
c) 8 5
d) 5 3

[View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int x;
3.  void main()
4.  {
5.      m();
6.      printf("%d", x);
7.  }
8.  void m()
9.  {
10.     x = 4;
11. }
```

a) 0

b) 4

c) Compile time error

d) Undefined

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  static int x = 5;
3.  void main()
4.  {
5.      int x = 9;
6.      {
7.          x = 4;
8.      }
9.      printf("%d", x);
10. }
```

- a) 9
- b) 5
- c) 4
- d) 0

View Answer

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      {
5.          int x = 8;
6.      }
7.      printf("%d", x);
8.  }
```

- a) 8
- b) 0
- c) Undefined
- d) Compile time error

View Answer

Answer: d

Explanation: None.

Static Variables – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      m();
5.      m();
6.  }
7.  void m()
8.  {
9.      static int x = 5;
```

```
10.    x++;  
11.    printf("%d", x);  
12.    }
```

- a) 6 7
- b) 6 6
- c) 5 5
- d) 5 6

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>  
2.    void main()  
3.    {  
4.        static int x;  
5.        printf("x is %d", x);  
6.    }
```

- a) 0
- b) 1
- c) Junk value
- d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>  
2.    static int x;  
3.    void main()  
4.    {  
5.        int x;  
6.        printf("x is %d", x);  
7.    }
```

- a) 0
- b) Junkvalue
- c) Run time error

d) Nothing

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      static double x;
5.      int x;
6.      printf("x is %d", x);
7.  }
```

a) Nothing

b) 0

c) Compile time error

d) Junkvalue

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      static int x;
5.      if (x++ < 2)
6.          main();
7.  }
```

a) Infinite calls to main

b) Run time error

c) Varies

d) main is called twice

[View Answer](#)

Answer: d

Explanation: None.

6. Which of following is not accepted in C?

a) static a = 10; //static as

- b) static int func (int); //parameter as static
- c) static static int a; //a static variable prefixed with static
- d) all of the mentioned

View Answer

Answer: c

Explanation: None.

7. Which of the following cannot be static in C?

- a) Variables
- b) Functions
- c) Structures
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

Static Variables – 2

1. What will be the output of the following C code if these two files namely test.c and test1.c are linked and run?

1. -----file test.c-----
2. #include <stdio.h>
3. #include ""test.h""
4. int main()
5. {
6. i = 10;
7. printf("""%d """, i);
8. foo();
9. }
- 10.
11. -----file test1.c-----
12. #include <stdio.h>
13. #include ""test.h""
14. int foo()
15. {
16. printf("""%d\n""", i);
17. }
- 18.

19. -----file test.h-----
20. #include <stdio.h>
21. #include <stdlib.h>
22. static int i;

- a) 10 0
- b) 0 0
- c) 10 10
- d) Compilation Error

View Answer

Answer: a

Explanation: None.

2. Functions have static qualifier for its declaration by default.

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

3. Is initialisation mandatory for local static variables?

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

1. #include <stdio.h>
2. int main()
3. {
4. foo();
5. foo();
6. }
7. void foo()
8. {
9. int i = 11;

```
10.    printf("%d ", i);  
11.    static int j = 12;  
12.    j = j + 1;  
13.    printf("%d\n", j);  
14. }
```

- a) 11 12 11 12
- b) 11 13 11 14
- c) 11 12 11 13
- d) Compile time error

View Answer

Answer: b

Explanation: None.

5. Assignment statements assigning value to local static variables are executed only once.

- a) True
- b) False
- c) Depends on the code
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

6. What is the format identifier for "static a = 20.5;"?

- a) %s
- b) %d
- c) %f
- d) Illegal declaration due to absence of data type

View Answer

Answer: b

Explanation: None.

7. Which of the following is true for the static variable?

- a) It can be called from another function
- b) It exists even after the function ends
- c) It can be modified in another function by sending it as a parameter
- d) All of the mentioned

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.    #include <stdio.h>  
2.    void func();
```

```

3.   int main()
4.   {
5.       static int b = 20;
6.       func();
7.   }
8.   void func()
9.   {
10.      static int b;
11.      printf("%d", b);
12.  }

```

- a) Output will be 0
- b) Output will be 20
- c) Output will be a garbage value
- d) Compile time error due to redeclaration of static variable

View Answer

Answer: a

Explanation: None.

Register Variables – 1

1. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int main()
3.   {
4.       register int i = 10;
5.       int *p = &i;
6.       *p = 11;
7.       printf("%d %d\n", i, *p);
8.   }

```

- a) Depends on whether i is actually stored in machine register
- b) 10 10
- c) 11 11
- d) Compile time error

View Answer

Answer: d

Explanation: None.

2. register keyword mandates compiler to place it in machine register.

- a) True
- b) False
- c) Depends on the standard
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      register static int i = 10;
5.      i = 11;
6.      printf("%d\n", i);
7.  }
```

- a) 10
- b) Compile time error
- c) Undefined behaviour
- d) 11

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      register auto int i = 10;
5.      i = 11;
6.      printf("%d\n", i);
7.  }
```

- a) 10
- b) Compile time error
- c) Undefined behaviour
- d) 11

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        register const int i = 10;
5.        i = 11;
6.        printf("%d\n", i);
7.    }
```

- a) 10
- b) Compile time error
- c) Undefined behaviour
- d) 11

View Answer

Answer: b

Explanation: None.

6. Register storage class can be specified to global variables.

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

7. Which among the following is wrong for "register int a;"?

- a) Compiler generally ignores the request
- b) You cannot take the address of this variable
- c) Access time to a is critical
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
```

```
4.     register int x = 5;
5.     m();
6.     printf("x is %d", x);
7. }
8. void m()
9. {
10.    x++;
11. }
```

- a) 6
- b) 5
- c) Junk value
- d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

Register Variables – 2

1. When compiler accepts the request to use the variable as a register?

- a) It is stored in CPU
- b) It is stored in cache memory
- c) It is stored in main memory
- d) It is stored in secondary memory

[View Answer](#)

Answer: a

Explanation: None.

2. Which data type can be stored in register?

- a) int
- b) long
- c) float
- d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

3. Which of the following operation is not possible in a register variable?

- a) Reading the value into a register variable
- b) Copy the value from a memory variable
- c) Global declaration of register variable
- d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

4. Which among the following is the correct syntax to declare a static variable register?

- a) static register a;
- b) register static a;
- c) Both static register a; and register static a;
- d) We cannot use static and register together

View Answer

Answer: d

Explanation: None.

5. Register variables reside in _____

- a) stack
- b) registers
- c) heap
- d) main memory

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  void main()
4.  {
5.      register int x = 0;
6.      if (x < 2)
7.      {
8.          x++;
9.      }
10. }
```

- a) Segmentation fault
- b) main is called twice
- c) main is called once
- d) main is called thrice

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?


```

1.    #include <stdio.h>
2.    void main()
3.    {
4.        register int x;
5.        printf("%d", x);
6.    }

```

- a) 0
- b) Junk value
- c) Compile time error
- d) Nothing

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    register int x;
3.    void main()
4.    {
5.        printf("%d", x);
6.    }

```

- a) Varies
- b) 0
- c) Junk value
- d) Compile time error

View Answer

Answer: d

Explanation: None.

Automatic Variables – 1

1. What is the scope of an automatic variable?

- a) Within the block it appears
- b) Within the blocks of the block it appears
- c) Until the end of program
- d) Within the block it appears & Within the blocks of the block it appears

View Answer

Answer: d

Explanation: None.

2. Automatic variables are allocated space in the form of a _____

- a) stack
- b) queue
- c) priority queue
- d) random

[View Answer](#)

Answer: a

Explanation: None.

3. Which of the following is a storage specifier?

- a) enum
- b) union
- c) auto
- d) volatile

[View Answer](#)

Answer: c

Explanation: None.

4. If storage class is not specified for a local variable, then the default class will be auto.

- a) True
- b) False
- c) Depends on the standard
- d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo(auto int i);
3.  int main()
4.  {
5.      foo(10);
6.  }
7.  void foo(auto int i)
8.  {
9.      printf("%d\n", i);
10. }
```

- a) 10
- b) Compile time error
- c) Depends on the standard

d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

6. Automatic variables are stored in _____

a) stack

b) data segment

c) register

d) heap

[View Answer](#)

Answer: a

Explanation: None.

7. What linkage does automatic variables have?

a) Internal linkage

b) External linkage

c) No linkage

d) None of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      auto i = 10;
5.      const auto int *p = &i;
6.      printf("%d\n", i);
7.  }
```

a) 10

b) Compile time error

c) Depends on the standard

d) Depends on the compiler

[View Answer](#)

Answer: a

Explanation: None.

Automatic Variables – 2

1. Automatic variables are _____

a) Declared within the scope of a block, usually a function

- b) Declared outside all functions
- c) Declared with the auto keyword
- d) Declared within the keyword extern

View Answer

Answer: a

Explanation: None.

2. What is the scope of an automatic variable?

- a) Exist only within that scope in which it is declared
- b) Cease to exist after the block is exited
- c) Exist only within that scope in which it is declared & exist after the block is exited
- d) All of the mentioned

View Answer

Answer: c

Explanation: None.

3. Automatic variables are allocated memory in _____

- a) heap
- b) Data segment
- c) Code segment
- d) stack

View Answer

Answer: d

Explanation: None.

4. What will be the x in the following C code?

1. `#include <stdio.h>`
2. `void main()`
3. `{`
4. `int x;`
5. `}`

- a) automatic variable
- b) static variable
- c) register variable
- d) global variable

View Answer

Answer: a

Explanation: None.

5. Automatic variables are initialized to _____

- a) Zero
- b) Junk value
- c) Nothing

d) Both Zero & Junk value

[View Answer](#)

Answer: b

Explanation: None.

6. Which of the following storage class supports char data type?

a) register

b) static

c) auto

d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

7. A local variable declaration with no storage class specified is by default _____

a) auto

b) extern

c) static

d) register

[View Answer](#)

Answer: a

Explanation: None.

C Preprocessor – 1

1. #include statement must be written _____

a) Before main()

b) Before any scanf/printf

c) After main()

d) It can be written anywhere

[View Answer](#)

Answer: d

Explanation: Even though using include directives before main() improves readability, it is not mandatory to write it before main() function. Only requirement is that we should include the header files before calling the library function.

2. #pragma exit is primarily used for?

a) Checking memory leaks after exiting the program

b) Informing Operating System that program has terminated

c) Running a function at exiting the program

d) No such preprocessor exist

[View Answer](#)

Answer: c

Explanation: It is primarily used for running a function upon exiting the program.

3. What will be the output of the following C code?

1. #include <stdio.h>

```

2.   int main()
3.   {
4.       int one = 1, two = 2;
5.       #ifdef next
6.           one = 2;
7.           two = 1;
8.       #endif
9.       printf("%d, %d", one, two);
10.  }

```

- a) 1, 1
- b) 1, 2
- c) 2, 1
- d) 2, 2

View Answer

Answer: b

Explanation: None.

4. What is #include directive?

- a) Tells the preprocessor to grab the text of a file and place it directly into the current file
- b) Statements are not typically placed at the top of a program
- c) All of the mentioned
- d) None of the mentioned

View Answer

Answer: a

Explanation: The #include directive tells the preprocessor to grab the text of a file and place it directly into the current file and are statements are typically placed at the top of a program.

5. The preprocessor provides the ability for _____

- a) The inclusion of header files
- b) The inclusion of macro expansions
- c) Conditional compilation and line control
- d) All of the mentioned

View Answer

Answer: d

Explanation: The preprocessor provides the ability for the inclusion of header files, macro expansions, conditional compilation, and line control.

6. If #include is used with file name in angular brackets.

- a) The file is searched for in the standard compiler include paths
- b) The search path is expanded to include the current source directory
- c) The search path will expand

d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: With the #include, if the filename is enclosed within angle brackets, the file is searched for in the standard compiler include paths.

C-Preprocessor - 2

1. What is a preprocessor?

- a) That processes its input data to produce output that is used as input to another program
- b) That is nothing but a loader
- c) That links various source files
- d) All of the mentioned

[View Answer](#)

Answer: a

Explanation: A preprocessor is a program that processes its input data to produce output that is used as input to another program.

2. Which of the following are C preprocessors?

- a) #ifdef
- b) #define
- c) #endif
- d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

3. Property which allows to produce different executable for different platforms in C is called?

- a) File inclusion
- b) Selective inclusion
- c) Conditional compilation
- d) Recursive macros

[View Answer](#)

Answer: c

Explanation: Conditional compilation is the preprocessor facility to produce a different executable.

4. What is #include <stdio.h>?

- a) Preprocessor directive
- b) Inclusion directive
- c) File inclusion directive
- d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

5. C preprocessors can have compiler specific features.

- a) True

- b) False
- c) Depends on the standard
- d) Depends on the platform

View Answer

Answer: a

Explanation: #pragma is compiler specific feature.

6. What will be the output of the following C code?

```
1. #include <stdio.h>
2. #define foo(m, n) m * n = 10
3. int main()
4. {
5.     printf("in main\n");
6. }
```

- a) In main
- b) Compilation error as lvalue is required for the expression m*n=10
- c) Preprocessor error as lvalue is required for the expression m*n=10
- d) None of the mentioned

View Answer

Answer: a

Explanation: Preprocessor just replaces whatever is given compiler then checks for error at the replaced part of the code. Here it is not replaced anywhere.

Output:

```
$ cc pgm1.c
```

```
$ a.out
```

```
in main
```

7. C preprocessor is conceptually the first step during compilation.

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

8. Preprocessor feature that supply line numbers and filenames to compiler is called?

- a) Selective inclusion
- b) macro substitution
- c) Concatenation
- d) Line control

View Answer

Answer: d

Explanation: None.

9. #include <somefile.h> are _____ files and #include "somefile.h" _____ files.

- a) Library, Library
- b) Library, user-created header
- c) User-created header, library
- d) They can include all types of file

View Answer

Answer: d

Explanation: Both of these statement can be used to select any file.

10. The C-preprocessors are specified with _____ symbol.

- a) #
- b) \$
- c) " "
- d) &

View Answer

Answer: a

Explanation: The C-preprocessors are specified with # symbol.

File Inclusion – 1

1. What is the sequence for preprocessor to look for the file within <>?

- a) The predefined location then the current directory
- b) The current directory then the predefined location
- c) The predefined location only
- d) The current directory location

View Answer

Answer: a

Explanation: <> first searches the predefined location for the specified file and then the current directory.

2. Which directory the compiler first looks for the file when using #include?

- a) Current directory where program is saved
- b) C:COMPILERINCLUDE
- c) S:SOURCEHEADERS
- d) Both C:COMPILERINCLUDE and S:SOURCEHEADERS simultaneously

View Answer

Answer: b

Explanation: None.

3. What would happen if you create a file stdio.h and use #include "stdio.h"?

- a) The predefined library file will be selected
- b) The user-defined library file will be selected
- c) Both the files will be included
- d) The compiler won't accept the program

View Answer

Answer: b

Explanation: None.

4. How is search done in #include and #include "somelibrary.h" according to C standard?

- a) When former is used, current directory is searched and when latter is used, standard directory is searched
- b) When former is used, standard directory is searched and when latter is used, current directory is searched
- c) When former is used, search is done in implementation defined manner and when latter is used, current directory is searched
- d) For both, search for 'somelibrary' is done in implementation-defined places

View Answer

Answer: b

Explanation: None.

5. How is search done in #include and #include "somelibrary.h" normally or conventionally?

- a) When former is used, current directory is searched and when latter is used, standard directory is searched
- b) When former is used, predefined directory is searched and when latter is used, current directory is searched and then predefined directories are searched
- c) When former is used, search is done in implementation defined manner and latter is used to search current directory
- d) For both, search for somelibrary is done in implementation-defined manner

View Answer

Answer: b

Explanation: None.

6. Can function definition be present in header files?

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

7. Comment on the output of the following C code.

```
1.  #include <stdio.h>
2.  #include "test.h"
3.  #include "test.h"
4.  int main()
5.  {
6.      //some code
7.  }
```

- a) True
- b) Compile time error
- c) False
- d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define foo(m, n) m ## n
3.  void myfunc();
4.  int main()
5.  {
6.      myfunc();
7.  }
8.  void myfunc()
9.  {
10.     printf("%d\n", foo(2, 3));
11. }
```

- a) 23
- b) 2 3
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

File Inclusion – 2

1. If the file name is enclosed in double quotation marks, then _____

- a) The preprocessor treats it as a user-defined file
- b) The preprocessor treats it as a system-defined file
- c) The preprocessor treats it as a user-defined file & system-defined file
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

2. If the file name is enclosed in angle brackets, then _____

- a) The preprocessor treats it as a user-defined file

- b) The preprocessor treats it as a system-defined file
- c) The preprocessor treats it as a user-defined file & system-defined file
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code snippet?

```
1.  #include (stdio.h)
2.  void main()
3.  {
4.      printf("hello");
5.  }
```

- a) hello
- b) Nothing
- c) Compile time error
- d) Depends on compiler

View Answer

Answer: c

Explanation: File to be included must be specified either in "" or <>.

Output:

```
$ cc pgm1.c
```

```
pgm1.c:1: error: #include expects "FILENAME" or
```

```
pgm1.c: In function 'main':
```

```
pgm1.c:4: warning: incompatible implicit declaration of built-in function 'printf'
```

4. The below two lines are equivalent to _____

```
1.  #define C_IO_HEADER <stdio.h>
2.  #include C_IO_HEADER
```

- a) #include<stdlib.h>
- b) #include"printf"
- c) #include"C_IO_HEADER"
- d) #include<stdio.h>

View Answer

Answer: d

Explanation: Since C_IO_HEADER is defined to be <stdio.h>, the second line becomes #include<stdio.h>, since C_IO_HEADER is replaced with <stdio.h>

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include "printf"
```

```
3. void main()
4. {
5.     printf("hello");
6. }
```

- a) hello
- b) Error
- c) Depends on compiler
- d) Varies

View Answer

Answer: b

Explanation: None.

6. Which of the following file extensions are accepted with #include?

- a) .h
- b) .in
- c) .com
- d) All of the mentioned

View Answer

Answer: d

Explanation: The preprocessor will include whatever file extension you specify in your #include statement. However, it is not a good practice as another person debugging it will find it difficult in finding files you have included.

7. Which of the following names for files not accepted?

- a) header.h.h
- b) 123header.h
- c) _head_er.h
- d) None of the mentioned

View Answer

Answer: d

Explanation: All file names are accepted as for the execution to occur. There are no constraints on giving file names for inclusion.

Macro Substitution – 1

1. What will be the output of the following C code?

```
1. #include <stdio.h>
2. #define foo(m, n) m ## n
3. int main()
4. {
5.     printf("%s\n", foo(k, l));
6. }
```

- a) k l
- b) kl
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    #define foo(m, n) " m ## n "
3.    int main()
4.    {
5.        printf("%s\n", foo(k, l));
6.    }
```

- a) k l
- b) kl
- c) Compile time error
- d) m ## n

View Answer

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    #define foo(x, y) #x #y
3.    int main()
4.    {
5.        printf("%s\n", foo(k, l));
6.        return 0;
7.    }
```

- a) kl
- b) k l
- c) xy
- d) Compile time error

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define foo(x, y) x / y + x
3.  int main()
4.  {
5.      int i = -6, j = 3;
6.      printf("%d\n",foo(i + j, 3));
7.      return 0;
8.  }
```

- a) Divided by zero exception
- b) Compile time error
- c) -8
- d) -4

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void f();
3.  int main()
4.  {
5.      #define foo(x, y) x / y + x
6.      f();
7.  }
8.  void f()
9.  {
10.     printf("%d\n", foo(-3, 3));
11. }
```

- a) -8
- b) -4
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void f();
3.  int main()
4.  {
5.      #define max 10
6.      f();
7.      return 0;
8.  }
9.  void f()
10. {
11.     printf("%d\n", max * 10);
12. }
```

- a) 100
- b) Compile time error since #define cannot be inside functions
- c) Compile time error since max is not visible in f()
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define foo(x, y) x / y + x
3.  int main()
4.  {
5.      int i = -6, j = 3;
6.      printf("%d ", foo(i + j, 3));
7.      printf("%d\n", foo(-3, 3));
8.      return 0;
9.  }
```


- a) -8 -4
- b) -4 divided by zero exception
- c) -4 -4
- d) Divided by zero exception

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int foo(int, int);
3.  #define foo(x, y) x / y + x
4.  int main()
5.  {
6.      int i = -6, j = 3;
7.      printf("%d ",foo(i + j, 3));
8.      #undef foo
9.      printf("%d\n",foo(i + j, 3));
10. }
11. int foo(int x, int y)
12. {
13.     return x / y + x;
14. }
```

- a) -8 -4
- b) Compile time error
- c) -8 -8
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

9. What is the advantage of #define over const?

- a) Data type is flexible
- b) Can have a pointer
- c) Reduction in the size of the program
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

Macro Substitution – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        #define max 37;
5.        printf("%d", max);
6.    }
```

- a) 37
 - b) Compile time error
 - c) Varies
 - d) Depends on compiler
- [View Answer](#)

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        #define max 37
5.        printf("%d", max);
6.    }
```

- a) 37
 - b) Run time error
 - c) Varies
 - d) Depends on compiler
- [View Answer](#)

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        #define const int
```

```
5.     const max = 32;
6.     printf("%d", max);
7. }
```

- a) Run time error
- b) 32
- c) int
- d) const

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         #define max 45
5.         max = 32;
6.         printf("%d", max);
7.     }
```

- a) 32
- b) 45
- c) Compile time error
- d) Varies

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     # define max
3.     void m()
4.     {
5.         printf("hi");
6.     }
7.     void main()
8.     {
```

9. max;
10. m();
11. }

- a) Run time error
- b) hi hi
- c) Nothing
- d) hi

[View Answer](#)

Answer: d

Explanation: None.

6. What will be the output of the following C code?

1. #include <stdio.h>
2. #define A 1 + 2
3. #define B 3 + 4
4. int main()
5. {
6. int var = A * B;
7. printf("%d\n", var);
8. }

- a) 9
- b) 11
- c) 12
- d) 21

[View Answer](#)

Answer: b

Explanation: None.

7. Which of the following Macro substitution are accepted in C?

a)

 #define A #define

 A VAR 20

b)

 #define A define

 #A VAR 20

c)

 #define #A #define

#A VAR 20

d) None of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

8. Comment on the output of the following C code.

```
1.    #include <stdio.h>
2.    #define var 20);
3.    int main()
4.    {
5.        printf("%d\n", var
6.    }
```

a) No errors, it will show the output 20

b) Compile time error, the printf braces aren't closed

c) Compile time error, there are no open braces in #define

d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

9. Which of the following properties of #define is not true?

a) You can use a pointer to #define

b) #define can be made externally available

c) They obey scope rules

d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

Conditional Inclusion – 1

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    #define SYSTEM 20
3.    int main()
4.    {
5.        int a = 20;
6.        #if SYSTEM == a
```

```
7.    printf("HELLO ");
8.    #endif
9.    #if SYSTEM == 20
10.   printf("WORLD\n");
11.   #endif
12. }
```

- a) HELLO
- b) WORLD
- c) HELLO WORLD
- d) No Output

[View Answer](#)

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    #define Cprog
3.    int main()
4.    {
5.        int a = 2;
6.        #ifdef Cprog
7.            a = 1;
8.            printf("%d", Cprog);
9.    }
```

- a) No output on execution
- b) Output as 1
- c) Output as 2
- d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

3. The “else if” in conditional inclusion is written by?

- a) #else if
- b) #elseif
- c) #elsif
- d) #elif

[View Answer](#)

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define COLD
3.  int main()
4.  {
5.      #ifdef COLD
6.          printf("COLD\t");
7.      #undef COLD
8.      #endif
9.      #ifdef COLD
10.         printf("HOT\t");
11.     #endif
12. }
```

- a) HOT
- b) COLD
- c) COLD HOT
- d) No Output

[View Answer](#)

Answer: b

Explanation: None.

5. Which of the following sequences are unaccepted in C language?

a)

```
#if
#else
#endif
```

b)

```
#if
#elif
#endif
```

c)

```
#if
```

```
#if  
#endif
```

d)

```
#if  
#undef  
#endif
```

View Answer

Answer: c

Explanation: None.

6. In a conditional inclusion, if the condition that comes after the if is true, then what will happen during compilation?

- a) Then the code up to the following #else or #elif or #endif is compiled
- b) Then the code up to the following #endif is compiled even if #else or #elif is present
- c) Then the code up to the following #elif is compiled
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

7. Conditional inclusion can be used for _____

- a) Preventing multiple declarations of a variable
- b) Check for existence of a variable and doing something if it exists
- c) Preventing multiple declarations of same function
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

8. The #elif directive cannot appear after the preprocessor #else directive.

- a) True
- b) False

View Answer

Answer: a

Explanation: None.

Conditional Inclusion – 2

1. For each #if, #ifdef, and #ifndef directive.

- a) There are zero or more #elif directives
- b) Zero or one #else directive
- c) One matching #endif directive
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

2. The #else directive is used for _____

- a) Conditionally include source text if the previous #if, #ifdef, #ifndef, or #elif test fails
- b) Conditionally include source text if a macro name is not defined
- c) Conditionally include source text if a macro name is defined
- d) Ending conditional text

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define MIN 0
3.  #if MIN
4.  #define MAX 10
5.  #endif
6.  int main()
7.  {
8.      printf("%d %d\n", MAX, MIN);
9.      return 0;
10. }
```

- a) 10 0
- b) Compile time error
- c) Undefined behaviour
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define MIN 0
3.  #ifdef MIN
4.  #define MAX 10
5.  #endif
6.  int main()
```

```

7.  {
8.      printf("%d %d\n", MAX, MIN);
9.      return 0;
10. }
```

- a) 10 0
- b) Compile time error
- c) Undefined behaviour
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  #define MIN 0
3.  #if defined(MIN) + defined(MAX)
4.  #define MAX 10
5.  #endif
6.  int main()
7.  {
8.      printf("%d %d\n", MAX, MIN);
9.      return 0;
10. }
```

- a) 10 0
- b) Compile time error
- c) Undefined behaviour
- d) Somegarbagevalue 0

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  #define MIN 0
3.  #if defined(MIN) - (!defined(MAX))
4.  #define MAX 10
```

```

5.  #endif
6.  int main()
7.  {
8.      printf("%d %d\n", MAX, MIN);
9.      return 0;
10. }

```

- a) 10 0
- b) Compile time error
- c) Undefined behaviour
- d) Somegarbagevalue 0

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  #define MIN 0
3.  #ifdef(MIN)
4.  #define MAX 10
5.  #endif
6.  int main()
7.  {
8.      printf("%d %d\n", MAX, MIN);
9.      return 0;
10. }

```

- a) 10 0
- b) Compile time error
- c) Run time error
- d) Preprocessor error

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  #define MIN 0);

```

```
3.  #ifdef MIN
4.  #define MAX 10
5.  #endif
6.  int main()
7.  {
8.      printf("%d %d\n", MAX, MIN
9.      return 0;
10. }
```

- a) 10 0
- b) Compile time error due to illegal syntax for printf
- c) Undefined behaviour
- d) Compile time error due to illegal MIN value

View Answer

Answer: a

Explanation: None.

Multiple Choice Questions on Pointers and Arrays in C

Pointers and Addresses - 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *p = NULL;
5.      char *q = 0;
6.      if (p)
7.          printf(" p ");
8.      else
9.          printf("nullp");
10.     if (q)
11.         printf("q\n");
12.     else
13.         printf(" nullq\n");
14. }
```

a) nullp nullq

b) Depends on the compiler

c) x nullq where x can be p or nullp depending on the value of NULL

d) p q

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 10;
5.      void *p = &i;
6.      printf("%d\n", (int)*p);
7.      return 0;
```

8. }

- a) Compile time error
- b) Segmentation fault/runtime crash
- c) 10
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4.     int i = 10;
5.     void *p = &i;
6.     printf("%f\n", *(float*)p);
7.     return 0;
8. }
```

- a) Compile time error
- b) Undefined behaviour
- c) 10
- d) 0.000000

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int *f();
3. int main()
4. {
5.     int *p = f();
6.     printf("%d\n", *p);
7. }
8. int *f()
9. {
```

```

10.    int *j = (int*)malloc(sizeof(int));
11.    *j = 10;
12.    return j;
13.    }

```

- a) 10
- b) Compile time error
- c) Segmentation fault/runtime crash since pointer to local variable is returned
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int *f();
3.    int main()
4.    {
5.        int *p = f();
6.        printf("%d\n", *p);
7.    }
8.    int *f()
9.    {
10.        int j = 10;
11.        return &j;
12.    }

```

- a) 10
- b) Compile time error
- c) Segmentation fault/runtime crash
- d) Undefined behaviour

View Answer

Answer: a

Explanation: We are returning address of a local variable which should not be done. In this specific instance, we are able to see the value of 10, which may not be the case if we call other functions before calling printf() in main().

6. Comment on the following pointer declaration.

```
int *ptr, p;
```

- a) ptr is a pointer to integer, p is not
- b) ptr and p, both are pointers to integer
- c) ptr is a pointer to integer, p may or may not be
- d) ptr and p both are not pointers to integer

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int *ptr, a = 10;
5.      ptr = &a;
6.      *ptr += 1;
7.      printf("%d,%d/n", *ptr, a);
8.  }
```

- a) 10,10
- b) 10,11
- c) 11,10
- d) 11,11

View Answer

Answer: d

Explanation: None.

8. Comment on the following C statement.

```
const int *ptr;
```

- a) You cannot change the value pointed by ptr
- b) You cannot change the pointer ptr itself
- c) You May or may not change the value pointed by ptr
- d) You can change the pointer as well as the value pointed by it

View Answer

Answer: a

Explanation: None.

Pointers and Addresses - 2

1. Which is an indirection operator among the following?

- a) &
- b) *
- c) ->

d) .

View Answer

Answer: b

Explanation: None.

2. Which of the following does not initialize ptr to null (assuming variable declaration of a as int a=0;)?

a) int *ptr = &a;

b) int *ptr = &a – &a;

c) int *ptr = a – a;

d) All of the mentioned

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int x = 0;
3.  void main()
4.  {
5.      int *ptr = &x;
6.      printf("%p\n", ptr);
7.      x++;
8.      printf("%p\n ", ptr);
9.  }
```

a) Same address

b) Different address

c) Compile time error

d) Varies

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int x = 0;
3.  void main()
4.  {
5.      int *const ptr = &x;
```

```

6.    printf("%p\n", ptr);
7.    ptr++;
8.    printf("%p\n ", ptr);
9.    }

```

- a) 0 1
- b) Compile time error
- c) 0xbfd605e8 0xbfd605ec
- d) 0xbfd605e8 0xbfd605e8

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    void main()
3.    {
4.        int x = 0;
5.        int *ptr = &x;
6.        printf("%p\n", ptr);
7.        ptr++;
8.        printf("%p\n ", ptr);
9.    }

```

- a) 0xbfd605e8 0xbfd605ec
- b) 0xbfd605e8 0cbfd60520
- c) 0xbfd605e8 0xbfd605e9
- d) Run time error

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    void main()
3.    {
4.        int x = 0;
5.        int *ptr = &5;

```

6. printf("%p\n", ptr);

7. }

a) 5

b) Address of 5

c) Nothing

d) Compile time error

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

1. #include <stdio.h>

2. void main()

3. {

4. int x = 0;

5. int *ptr = &x;

6. printf("%d\n", *ptr);

7. }

a) Address of x

b) Junk value

c) 0

d) Run time error

View Answer

Answer: c

Explanation: None.

Pointers and Function Arguments – 1

1. What will be the output of the following C code?

1. #include <stdio.h>

2. void foo(int*);

3. int main()

4. {

5. int i = 10;

6. foo(&i++);

7. }

8. void foo(int *p)

```
9.  {
10.    printf("%d\n", *p);
11. }
```

- a) 10
- b) Some garbage value
- c) Compile time error
- d) Segmentation fault/code crash

[View Answer](#)

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo(int*);
3.  int main()
4.  {
5.      int i = 10, *p = &i;
6.      foo(p++);
7.  }
8.  void foo(int *p)
9.  {
10.     printf("%d\n", *p);
11. }
```

- a) 10
- b) Some garbage value
- c) Compile time error
- d) Segmentation fault

[View Answer](#)

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo(float *);
3.  int main()
4.  {
```

```

5.     int i = 10, *p = &i;
6.     foo(&i);
7. }
8. void foo(float *p)
9. {
10.    printf("%f\n", *p);
11. }

```

- a) 10.000000
 - b) 0.000000
 - c) Compile time error
 - d) Undefined behaviour
- [View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 97, *p = &i;
5.      foo(&i);
6.      printf("%d ", *p);
7.  }
8.  void foo(int *p)
9.  {
10.     int j = 2;
11.     p = &j;
12.     printf("%d ", *p);
13. }

```

- a) 2 97
 - b) 2 2
 - c) Compile time error
 - d) Segmentation fault/code crash
- [View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 97, *p = &i;
5.      foo(&p);
6.      printf("%d ", *p);
7.      return 0;
8.  }
9.  void foo(int **p)
10. {
11.     int j = 2;
12.     *p = &j;
13.     printf("%d ", **p);
14. }
```

a) 2 2

b) 2 97

c) Undefined behaviour

d) Compilation Error

[View Answer](#)

Answer: c

Explanation: The main() function calls foo(&p) passing the address of an integer pointer. Inside foo() function, we are assigning the address of a local variable j to the pointer (*p) and then printing **p, which will display the value 2. However, once we return back to the caller, i.e., the main() function, we are trying to access the address of the local variable j, which was on another stack frame and it might have got destroyed once we returned back to the caller. If the memory for the stack frame of foo() function was still there, the program will print the value 2 in the main() function also. However, if the memory was not there, it will be invalid memory access, resulting in segmentation fault and maybe, a core dump. So, the correct answer will be "Undefined behaviour".

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 11;
```

```

5.     int *p = &i;
6.     foo(&p);
7.     printf("%d ", *p);
8. }
9. void foo(int *const *p)
10. {
11.     int j = 10;
12.     *p = &j;
13.     printf("%d ", **p);
14. }

```

- a) Compile time error
- b) 10 10
- c) Undefined behaviour
- d) 10 11

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void foo(int *);
3.     int main()
4.     {
5.         int i = 10;
6.         int *p = &i;
7.         foo(p);
8.         printf("%d ", *p);
9.     }
10. void foo(int *p)
11. {
12.     int j = 11;
13.     p = &j;
14.     printf("%d ", *p);
15. }

```

- a) 11 11
- b) 11 10
- c) Compile time error
- d) Undefined-value

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 10;
5.      int *const p = &i;
6.      foo(&p);
7.      printf("%d\n", *p);
8.  }
9.  void foo(int **p)
10. {
11.     int j = 11;
12.     *p = &j;
13.     printf("%d\n", **p);
14. }
```

- a) 11 11
- b) 11 10
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: d

Explanation: p points to invalid memory(local variable of another function) after it returns from foo() function.

9. Which of the following is the correct syntax to send an array as a parameter to function?

- a) func(&array);
- b) func(#array);
- c) func(*array);
- d) func(array[size]);

View Answer

Answer: a

Explanation: None.

Pointers and Function Arguments – 2

1. Which of the following can never be sent by call-by-value?

- a) Variable
- b) Array
- c) Structures
- d) Both Array and Structures

View Answer

Answer: b

Explanation: None.

2. Which type of variables can have the same name in a different function?

- a) Global variables
- b) Static variables
- c) Function arguments
- d) Both static variables and Function arguments

View Answer

Answer: d

Explanation: None.

3. Arguments that take input by user before running a program are called?

- a) Main function arguments
- b) Main arguments
- c) Command-Line arguments
- d) Parameterized arguments

View Answer

Answer: c

Explanation: None.

4. What is the maximum number of arguments that can be passed in a single function?

- a) 127
- b) 253
- c) 361
- d) No limits in number of arguments

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

- 1. `#include <stdio.h>`
- 2. `void m(int *p, int *q)`
- 3. `{`
- 4. `int temp = *p; *p = *q; *q = temp;`

```

5.     }
6.     void main()
7.     {
8.         int a = 6, b = 5;
9.         m(&a, &b);
10.        printf("%d %d\n", a, b);
11.    }

```

- a) 5 6
- b) 6 5
- c) 5 5
- d) 6 6

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void m(int *p)
3.     {
4.         int i = 0;
5.         for(i = 0; i < 5; i++)
6.             printf("%d\t", p[i]);
7.     }
8.     void main()
9.     {
10.        int a[5] = {6, 5, 3};
11.        m(&a);
12.    }

```

- a) 0 0 0 0 0
- b) 6 5 3 0 0
- c) Run time error
- d) 6 5 3 junk junk

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void m(int p, int q)
3.  {
4.      int temp = p;
5.      p = q;
6.      q = temp;
7.  }
8.  void main()
9.  {
10.     int a = 6, b = 5;
11.     m(a, b);
12.     printf("%d %d\n", a, b);
13. }
```

a) 5 6

b) 5 5

c) 6 5

d) 6 6

[View Answer](#)

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void m(int p, int q)
3.  {
4.      printf("%d %d\n", p, q);
5.  }
6.  void main()
7.  {
8.      int a = 6, b = 5;
9.      m(a);
10. }
```

- a) 6
- b) 6 5
- c) 6 junk value
- d) Compile time error

View Answer

Answer: d

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void m(int p)
3.  {
4.      printf("%d\n", p);
5.  }
6.  void main()
7.  {
8.      int a = 6, b = 5;
9.      m(a, b);
10.     printf("%d %d\n", a, b);
11. }
```

- a) 6
- b) 6 5
- c) 6 junk value
- d) Compile time error

View Answer

Answer: d

Explanation: None.

Pointers and Arrays - 1

1. What will be output of the following C code where we copy an array 'a' into array 'b' and then the array 'b' into 'a'?

```
#include<stdio.h>
#include<string.h>

main()
{
    char a[] = "hell";
    char b[] = "hello";
```

```
strcpy(b, a);  
strcpy(a, b);  
printf("%s, %s", a, b);  
}
```

a)

hello, hello

b)

hell, hell

c)

hell, hello

d) Runtime Error

[View Answer](#)

Answer: b

Explanation: In the above code, we are copying first the content of array a into array b. So, "hell" gets copied to array b. After that we are copying the content of array b into array a. Since, array b contained the string "hell" now, it gets copied to array a. Hence, the final answer will be "hell, hell".

2. Which keyword is used to make the array size optional in C language during array declaration?

a) auto

b) static

c) extern

d) register

[View Answer](#)

Answer: c

Explanation: None.

3. Which of the following is the correct syntax to send an array as a parameter to a C function?

a) Both func(&array) and func(*array);

b) Both func(#array) and func(&array);

c) Both func(array) and func(&array);

d) Both func(array[size]) and func(*array);

[View Answer](#)

Answer: c

Explanation: None.

4. What are the different ways to initialize an array with all elements as zero?

a) int array[5] = {};

b) int array[5] = {0};

c)

int a = 0, b = 0, c = 0;

```
int array[5] = {a, b, c};
```

d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

5. What are the elements present in the array of the following C code?

```
int array[5] = {5};
```

a) 5, 5, 5, 5, 5

b) 5, 0, 0, 0, 0

c) 5, (garbage), (garbage), (garbage), (garbage)

d) (garbage), (garbage), (garbage), (garbage), 5

[View Answer](#)

Answer: b

Explanation: None.

6. Which of the following declaration is illegal?

a)

```
int a = 0, b = 1, c = 2;
```

```
int array[3] = {a, b, c};
```

b)

```
int size = 3;
```

```
int array[size];
```

c)

```
int size = 3;
```

```
int array[size] = {1, 2, 3};
```

d) All of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

7. An array of similar data types which themselves are a collection of dissimilar data type are

a) Linked Lists

b) Trees

c) Array of Structure

d) All of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

8. Comment on an array of the void data type.

- a) It can store any data-type
- b) It only stores element of similar data type to first element
- c) It acquires the data type with the highest precision in it
- d) You cannot have an array of void data type

View Answer

Answer: d

Explanation: None.

9. An array in C cannot be initialized by which of the following statement?

- a) `char a[] = "Hello";`
- b) `char a[6] = {};`
- c) `char a[6] = {0};`
- d)

`char a[6];`

`a = "Hello";`

View Answer

Answer: d

Explanation: None.

10. What is the data type of the array passed from the command line into the `main()` function in C?

- a) `char arr[];`
- b) `char *arr[];`
- c) `char **arr[];`
- d) `char arr[][];`

View Answer

Answer: b

Explanation: None.

11. Which of the following c statement will calculate the correct size of an array of 10 integers?

(Assuming the declaration as `int a[10];`)

- a) `sizeof(a[10]);`
- b) `sizeof(*a);`
- c) `sizeof(a);`
- d) `sizeof(&a);`

View Answer

Answer: c

Explanation: None.

Pointers and Arrays - 2

1. What will be the output of the following C code?

1. `#include <stdio.h>`
2. `void main()`

```

3.  {
4.      int a[3] = {1, 2, 3};
5.      int *p = a;
6.      printf("%p\t%p", p, a);
7.  }

```

- a) Same address is printed
- b) Different address is printed
- c) Compile time error
- d) Nothing

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *s = "hello";
5.      char *p = s;
6.      printf("%p\t%p", p, s);
7.  }

```

- a) Different address is printed
- b) Same address is printed
- c) Run time error
- d) Nothing

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *s= "hello";
5.      char *p = s;
6.      printf("%c\t%c", p[0], s[1]);

```


7. }

- a) Run time error
- b) h h
- c) h e
- d) h l

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1. #include <stdio.h>
2. void main()
3. {
4.     char *s= "hello";
5.     char *p = s;
6.     printf("%c\t%c", *(p + 3), s[1]);
7. }
```

- a) h e
- b) l l
- c) l o
- d) l e

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1. #include <stdio.h>
2. void main()
3. {
4.     char *s= "hello";
5.     char *p = s;
6.     printf("%c\t%c", 1[p], s[1]);
7. }
```

- a) h h
- b) Run time error
- c) l l
- d) e e

View Answer

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo( int[] );
3.  int main()
4.  {
5.      int ary[4] = {1, 2, 3, 4};
6.      foo(ary);
7.      printf("%d ", ary[0]);
8.  }
9.  void foo(int p[4])
10. {
11.     int i = 10;
12.     p = &i;
13.     printf("%d ", p[0]);
14. }
```

a) 10 10

b) Compile time error

c) 10 1

d) Undefined behaviour

[View Answer](#)

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int ary[4] = {1, 2, 3, 4};
5.      int *p = ary + 3;
6.      printf("%d\n", p[-2]);
7.  }
```

- a) 1
- b) 2
- c) Compile time error
- d) Some garbage value

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int ary[4] = {1, 2, 3, 4};
5.        int *p = ary + 3;
6.        printf("%d %d\n", p[-2], ary[*p]);
7.    }
```

- a) 2 3
- b) Compile time error
- c) 2 4
- d) 2 somegarbagevalue

View Answer

Answer: d

Explanation: None.

9. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int ary[4] = {1, 2, 3, 4};
5.        printf("%d\n", *ary);
6.    }
```

- a) 1
- b) Compile time error
- c) Some garbage value
- d) Undefined variable

View Answer

Answer: a

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      const int ary[4] = {1, 2, 3, 4};
5.      int *p;
6.      p = ary + 3;
7.      *p = 5;
8.      printf("%d\n", ary[3]);
9.  }
```

- a) 4
- b) 5
- c) Compile time error
- d) 3

[View Answer](#)

Answer: b

Explanation: None.

11. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int ary[4] = {1, 2, 3, 4};
5.      int p[4];
6.      p = ary;
7.      printf("%d\n", p[1]);
8.  }
```

- a) 1
- b) Compile time error
- c) Undefined behaviour
- d) 2

[View Answer](#)

Answer: b

Explanation: None.

Address Arithmetic – 1

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        double *ptr = (double *)100;
5.        ptr = ptr + 2;
6.        printf("%u", ptr);
7.    }
```

- a) 102
- b) 104
- c) 108
- d) 116

View Answer

Answer: d

Explanation:None

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int *p = (int *)2;
5.        int *q = (int *)3;
6.        printf("%d", p + q);
7.    }
```

- a) 2
- b) 3
- c) 5
- d) Compile time error

View Answer

Answer: d

Explanation: None.

3. Which of the following arithmetic operation can be applied to pointers a and b?
(Assuming initialization as `int *a = (int *)2; int *b = (int *)3;`)

- a) $a + b$
- b) $a - b$
- c) $a * b$

d) a / b

View Answer

Answer: b

Explanation: None.

4. What is the size of *ptr in a 32-bit machine (Assuming initialization as int *ptr = 10;)?

a) 1

b) 2

c) 4

d) 8

View Answer

Answer: c

Explanation: None.

5. Which of following logical operation can be applied to pointers?

(Assuming initialization int *a = 2; int *b = 3;)

a) a | b

b) a ^ b

c) a & b

d) None of the mentioned

View Answer

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *s = "hello";
5.      char *p = s;
6.      printf("%c\t%c", *(p + 1), s[1]);
7.  }
```

a) h e

b) e l

c) h h

d) e e

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```

2. void main()
3. {
4.     char *s = "hello";
5.     char *p = s;
6.     printf("%c\t%c", *p, s[1]);
7. }

```

- a) e h
- b) Compile time error
- c) h h
- d) h e

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```

1. #include <stdio.h>
2. void main()
3. {
4.     char *s = "hello";
5.     char *n = "cjh";
6.     char *p = s + n;
7.     printf("%c\t%c", *p, s[1]);
8. }

```

- a) h e
- b) Compile time error
- c) c o
- d) h n

View Answer

Answer: b

Explanation: None.

Address Arithmetic – 2

1. What will be the output of the following C code?

```

1. #include <stdio.h>
2. void main()
3. {

```

```

4.     char *s = "hello";
5.     char *p = s * 3;
6.     printf("%c\t%c", *p, s[1]);
7.     }

```

- a) h e
- b) l e
- c) Compile time error
- d) l h

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void main()
3.     {
4.         char *s= "hello";
5.         char *p = s + 2;
6.         printf("%c\t%c", *p, s[1]);
7.     }

```

- a) l e
- b) h e
- c) l l
- d) h l

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         void *p;
5.         int a[4] = {1, 2, 3, 8};
6.         p = &a[3];
7.         int *ptr = &a[2];

```



```
8.     int n = p - ptr;
9.     printf("%d\n", n);
10. }
```

- a) 1
- b) Compile time error
- c) Segmentation fault
- d) 4

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         void *p;
5.         int a[4] = {1, 2, 3, 4};
6.         p = &a[3];
7.         int *ptr = &a[2];
8.         int n = (int*)p - ptr;
9.         printf("%d\n", n);
10. }
```

- a) 1
- b) Compile time error
- c) Segmentation fault
- d) 4

[View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int a[4] = {1, 2, 3, 4};
5.         int b[4] = {1, 2, 3, 4};
```

```
6.     int n = &b[3] - &a[2];
7.     printf("%d\n", n);
8. }
```

- a) -3
 - b) 5
 - c) 4
 - d) Can't say
- [View Answer](#)

Answer: d

Explanation: It depends on how the spaces for local variables are allocated on the runtime stack by the compiler. In some compiler, the answer will be 5, whereas in other compiler the answer will be -3. So, it is compiler dependent.

6. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int a[4] = {1, 2, 3, 4};
5.         int *p = &a[1];
6.         int *ptr = &a[2];
7.         ptr = ptr * 1;
8.         printf("%d\n", *ptr);
9.     }
```

- a) 2
 - b) 1
 - c) Compile time error
 - d) Undefined behaviour
- [View Answer](#)

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         int a[4] = {1, 2, 3, 4};
5.         int *ptr = &a[2];
```

```

6.    float n = 1;
7.    ptr = ptr + n;
8.    printf("%d\n", *ptr);
9.    }

```

- a) 4
- b) 3
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        int a[4] = {1, 2, 3, 4};
5.        void *p = &a[1];
6.        void *ptr = &a[2];
7.        int n = 1;
8.        n = ptr - p;
9.        printf("%d\n", n);
10.   }

```

- a) 1
- b) 4
- c) Compile time error
- d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

Character Pointers and Functions – 1

1. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {

```

```

4.     char *str = "hello, world\n";
5.     char *strc = "good morning\n";
6.     strcpy(strc, str);
7.     printf("%s\n", strc);
8.     return 0;
9.     }

```

- a) hello, world
- b) Crash/segmentation fault
- c) Undefined behaviour
- d) Run time error

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         char *str = "hello world";
5.         char strc[] = "good morning india\n";
6.         strcpy(strc, str);
7.         printf("%s\n", strc);
8.         return 0;
9.     }

```

- a) hello world
- b) hello worldg india
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {

```

```

4.     char *str = "hello, world!!\n";
5.     char strc[] = "good morning\n";
6.     strcpy(strc, str);
7.     printf("%s\n", strc);
8.     return 0;
9.     }

```

- a) hello, world!!
- b) Compile time error
- c) Undefined behaviour
- d) Segmentation fault

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         char *str = "hello, world\n";
5.         str[5] = '.';
6.         printf("%s\n", str);
7.         return 0;
8.     }

```

- a) hello. world
- b) hello, world
- c) Compile error
- d) Segmentation fault

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         char str[] = "hello, world";

```

```
5.     str[5] = '.';
6.     printf("%s\n", str);
7.     return 0;
8. }
```

- a) hello. world
- b) hello, world
- c) Compile error
- d) Segmentation fault

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         char *str = "hello world";
5.         char strary[] = "hello world";
6.         printf("%d %d\n", sizeof(str), sizeof(strary));
7.         return 0;
8.     }
```

- a) 11 11
- b) 12 12
- c) 4 12
- d) 4 11

View Answer

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     int main()
3.     {
4.         char *str = "hello world";
5.         char strary[] = "hello world";
6.         printf("%d %d\n", strlen(str), strlen(strary));
```

7. return 0;

8. }

a) 11 11

b) 12 11

c) 11 12

d) x 11 where x can be any positive integer.

[View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code?

1. #include <stdio.h>

2. void f(char *k)

3. {

4. k++;

5. k[2] = 'm';

6. printf("%c\n", *k);

7. }

8. void main()

9. {

10. char s[] = "hello";

11. f(s);

12. }

a) l

b) e

c) h

d) o

[View Answer](#)

Answer: b

Explanation: None.

9. What will be the output of the following C code?

1. #include <stdio.h>

2. void fun(char *k)

3. {

4. printf("%s", k);

```

5.    }
6.    void main()
7.    {
8.        char s[] = "hello";
9.        fun(s);
10.   }

```

- a) hello
- b) Run time error
- c) Nothing
- d) h

View Answer

Answer: a

Explanation: None.

Character Pointers and Functions – 2

1. Comment on the output of the following C code.

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        char *str = "This" //Line 1
5.        char *ptr = "Program\n"; //Line 2
6.        str = ptr; //Line 3
7.        printf("%s, %s\n", str, ptr); //Line 4
8.    }

```

- a) Memory holding “this” is cleared at line 3
- b) Memory holding “this” loses its reference at line 3
- c) You cannot assign pointer like in Line 3
- d) Output will be This, Program

View Answer

Answer: b

Explanation: None.

2. What type of initialization is needed for the segment “ptr[3] = ‘3’;” to work?

- a) char *ptr = “Hello!”;
- b) char ptr[] = “Hello!”;
- c) both char *ptr = “Hello!”; and char ptr[] = “Hello!”;
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

3. What is the syntax for constant pointer to address (i.e., fixed pointer address)?

- a) `const <type> * <name>`
- b) `<type> * const <name>`
- c) `<type> const * <name>`
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int add(int a, int b)
3.  {
4.      return a + b;
5.  }
6.  int main()
7.  {
8.      int (*fn_ptr)(int, int);
9.      fn_ptr = add;
10.     printf("The sum of two numbers is: %d", (int)fn_ptr(2, 3));
11. }
```

- a) Compile time error, declaration of a function inside main
- b) Compile time error, no definition of function `fn_ptr`
- c) Compile time error, illegal application of statement `fn_ptr = add`
- d) No Run time error, output is 5

View Answer

Answer: d

Explanation: None.

5. What is the correct way to declare and assign a function pointer?

(Assuming the function to be assigned is `"int multi(int, int);"`)

- a) `int (*fn_ptr)(int, int) = multi;`
- b) `int *fn_ptr(int, int) = multi;`
- c) `int *fn_ptr(int, int) = &multi;`
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

6. Calling a function f with a an array variable a[3] where a is an array, is equivalent to _____

- a) f(a[3])
- b) f(*(a + 3))
- c) f(3[a])
- d) all of the mentioned

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void f(char *k)
3.  {
4.      k++;
5.      k[2] = 'm';
6.  }
7.  void main()
8.  {
9.      char s[] = "hello";
10.     f(s);
11.     printf("%c\n", *s);
12. }
```

- a) h
- b) e
- c) m
- d) o;

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char s[] = "hello";
```

```
5.     s++;  
6.     printf("%c\n", *s);  
7. }
```

- a) Compile time error
- b) h
- c) e
- d) o

[View Answer](#)

Answer: a

Explanation: None.

Pointers to Pointers – 1

1. What will be the output of the following C code?

```
1.     #include <stdio.h>  
2.     void main()  
3.     {  
4.         int k = 5;  
5.         int *p = &k;  
6.         int **m = &p;  
7.         printf("%d%d%d\n", k, *p, **m);  
8.     }
```

- a) 5 5 5
- b) 5 5 junk value
- c) 5 junk junk
- d) Run time error

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.     #include <stdio.h>  
2.     void main()  
3.     {  
4.         int k = 5;  
5.         int *p = &k;  
6.         int **m = &p;
```

```
7.     printf("%d%d%d\n", k, *p, **p);  
8. }
```

- a) 5 5 5
- b) 5 5 junk value
- c) 5 junk junk
- d) Compile time error

View Answer

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
1.     #include <stdio.h>  
2.     void main()  
3.     {  
4.         int k = 5;  
5.         int *p = &k;  
6.         int **m = &p;  
7.         **m = 6;  
8.         printf("%d\n", k);  
9.     }
```

- a) 5
- b) Compile time error
- c) 6
- d) Junk

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.     #include <stdio.h>  
2.     void main()  
3.     {  
4.         int a[3] = {1, 2, 3};  
5.         int *p = a;  
6.         int *r = &p;  
7.         printf("%d", (**r));
```

8. }

- a) 1
- b) Compile time error
- c) Address of a
- d) Junk value

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int a[3] = {1, 2, 3};
5.      int *p = a;
6.      int **r = &p;
7.      printf("%p %p", *r, a);
8.  }
```

- a) Different address is printed
- b) 1 2
- c) Same address is printed
- d) 1 1

View Answer

Answer: c

Explanation: None.

6. How many number of pointer (*) does C have against a pointer variable declaration?

- a) 7
- b) 127
- c) 255
- d) No limits

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 1, b = 2, c = 3;
```

```

5.     int *ptr1 = &a, *ptr2 = &b, *ptr3 = &c;
6.     int **sptr = &ptr1; //-Ref
7.     *sptr = ptr2;
8.     }

```

- a) ptr1 points to a
- b) ptr1 points to b
- c) sptr points to ptr2
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void main()
3.     {
4.         int a[3] = {1, 2, 3};
5.         int *p = a;
6.         int **r = &p;
7.         printf("%p %p", *r, a);
8.     }

```

- a) Different address is printed
- b) 1 2
- c) Same address is printed
- d) 1 1

View Answer

Answer: c

Explanation: None.

Pointers to Pointers – 2

1. What substitution should be made to //-Ref such that ptr1 points to variable c in the following C code?

```

1.     #include <stdio.h>
2.     int main()
3.     {
4.         int a = 1, b = 2, c = 3;
5.         int *ptr1 = &a;

```

6. int **sptr = &ptr1;

7. // -Ref

8. }

a) *sptr = &c;

b) **sptr = &c;

c) *ptr1 = &c;

d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

2. Which of the following declaration will result in run-time error?

a) int **c = &c;

b) int **c = &*c;

c) int **c = **c;

d) none of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

3. Comment on the output of the following C code.

1. #include <stdio.h>

2. int main()

3. {

4. int a = 10;

5. int **c -= &&a;

6. }

a) You cannot apply any arithmetic operand to a pointer

b) We don't have address of an address operator

c) We have address of an address operator

d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

1. #include <stdio.h>

2. void main()

3. {

4. int k = 5;

```
5.     int *p = &k;
6.     int **m = &p;
7.     printf("%d%d%d\n", k, *p, **m);
8.     }
```

- a) 5 5 5
- b) 5 5 junk value
- c) 5 junk junk
- d) Compile time error

[View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         int k = 5;
5.         int *p = &k;
6.         int **m = &p;
7.         printf("%d%d%d\n", k, *p, **p);
8.     }
```

- a) 5 5 5
- b) 5 5 junk value
- c) 5 junk junk
- d) Compile time error

[View Answer](#)

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         int k = 5;
5.         int *p = &k;
6.         int **m = &p;
```



```
7.     **m = 6;

8.     printf("%d\n", k);

9.     }
```

- a) 5
 - b) Run time error
 - c) 6
 - d) Junk
- [View Answer](#)

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.     #include <stdio.h>

2.     void main()

3.     {

4.         int a[3] = {1, 2, 3};

5.         int *p = a;

6.         int *r = &p;

7.         printf("%d", (**r));

8.     }
```

- a) 1
 - b) Compile time error
 - c) Address of a
 - d) Junk value
- [View Answer](#)

Answer: b

Explanation: None.

Multidimensional Arrays – 1

1. What will be the output of the following C code?

```
1.     #include <stdio.h>

2.     void main()

3.     {

4.         int a[2][3] = {1, 2, 3, 4, 5};

5.         int i = 0, j = 0;

6.         for (i = 0; i < 2; i++)
```

```

7.     for (j = 0; j < 3; j++)
8.     printf("%d", a[i][j]);
9.     }

```

- a) 1 2 3 4 5 0
- b) 1 2 3 4 5 junk
- c) 1 2 3 4 5 5
- d) Run time error

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void main()
3.     {
4.         int a[2][3] = {1, 2, 3, , 4, 5};
5.         int i = 0, j = 0;
6.         for (i = 0; i < 2; i++)
7.         for (j = 0; j < 3; j++)
8.         printf("%d", a[i][j]);
9.     }

```

- a) 1 2 3 junk 4 5
- b) Compile time error
- c) 1 2 3 0 4 5
- d) 1 2 3 3 4 5

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     void f(int a[][3])
3.     {
4.         a[0][1] = 3;
5.         int i = 0, j = 0;
6.         for (i = 0; i < 2; i++)

```

```

7.     for (j = 0; j < 3; j++)
8.     printf("%d", a[i][j]);
9.     }
10.    void main()
11.    {
12.        int a[2][3] = {0};
13.        f(a);
14.    }

```

- a) 0 3 0 0 0 0
- b) Junk 3 junk junk junk junk
- c) Compile time error
- d) All junk values

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    void f(int a[][])
3.    {
4.        a[0][1] = 3;
5.        int i = 0, j = 0;
6.        for (i = 0; i < 2; i++)
7.            for (j = 0; j < 3; j++)
8.                printf("%d", a[i][j]);
9.    }
10.   void main()
11.   {
12.       int a[2][3] = {0};
13.       f(a);
14.   }

```

- a) 0 3 0 0 0 0
- b) Junk 3 junk junk junk junk
- c) Compile time error

d) All junk values

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void f(int a[2][])
3.  {
4.      a[0][1] = 3;
5.      int i = 0, j = 0;
6.      for (i = 0; i < 2; i++)
7.          for (j = 0; j < 3; j++)
8.              printf("%d", a[i][j]);
9.  }
10. void main()
11. {
12.     int a[2][3] = {0};
13.     f(a);
14. }
```

a) 0 3 0 0 0 0

b) Junk 3 junk junk junk junk

c) Compile time error

d) All junk values

View Answer

Answer: c

Explanation: None.

6. Comment on the following C statement.

```
int (*a)[7];
```

a) An array "a" of pointers

b) A pointer "a" to an array

c) A ragged array

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

7. Comment on the following 2 arrays with respect to P and Q.

1. `int *a1[8];`
2. `int *(a2[8]);`
3. P. Array of pointers
4. Q. Pointer to an array

- a) a1 is P, a2 is Q
- b) a1 is P, a2 is P
- c) a1 is Q, a2 is P
- d) a1 is Q, a2 is Q

View Answer

Answer: b

Explanation: None.

8. Which of the following is not possible statically in C?

- a) Jagged Array
- b) Rectangular Array
- c) Cuboidal Array
- d) Multidimensional Array

View Answer

Answer: a

Explanation: None.

Multidimensional Arrays – 2

1. What is the correct syntax to send a 3-dimensional array as a parameter? (Assuming declaration `int a[5][4][3];`)

- a) `func(a);`
- b) `func(&a);`
- c) `func(*a);`
- d) `func(**a);`

View Answer

Answer: a

Explanation: None.

2. What are the applications of a multidimensional array?

- a) Matrix-Multiplication
- b) Minimum Spanning Tree
- c) Finding connectivity between nodes
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void foo(int *ary[]);
3.  int main()
4.  {
5.      int ary[2][3];
6.      foo(ary);
7.  }
8.  void foo(int *ary[])
9.  {
10.     int i = 10, j = 2, k;
11.     ary[0] = &i;
12.     ary[1] = &j;
13.     *ary[0] = 2;
14.     for (k = 0; k < 2; k++)
15.         printf("%d\n", *ary[k]);
16. }

```

- a) 2 2
- b) Compile time error
- c) Undefined behaviour
- d) 10 2

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void foo(int (*ary)[3]);
3.  int main()
4.  {
5.      int ary[2][3];
6.      foo(ary);
7.  }
8.  void foo(int (*ary)[3])
9.  {

```

```
10.    int i = 10, j = 2, k;  
11.    ary[0] = &i;  
12.    ary[1] = &j;  
13.    for (k = 0; k < 2; k++)  
14.    printf("%d\n", *ary[k]);  
15. }
```

- a) Compile time error
- b) 10 2
- c) Undefined behaviour
- d) segmentation fault/code crash

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.    #include <stdio.h>  
2.    int main()  
3.    {  
4.        foo(ary);  
5.    }  
6.    void foo(int **ary)  
7.    {  
8.        int i = 10, k = 20, j = 30;  
9.        int *ary[2];  
10.     ary[0] = &i;  
11.     ary[1] = &j;  
12.     printf("%d\n", ary[0][1]);  
13. }
```

- a) 10
- b) 20
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int ary[2][3][4], j = 20;
5.        ary[0][0] = &j;
6.        printf("%d\n", *ary[0][0]);
7.    }
```

- a) Compile time error
- b) 20
- c) Address of j
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int ary[2][3];
5.        ary[][] = {{1, 2, 3}, {4, 5, 6}};
6.        printf("%d\n", ary[1][0]);
7.    }
```

- a) Compile time error
- b) 4
- c) 1
- d) 2

View Answer

Answer: a

Explanation: None.

Initialization of Pointer Arrays – 1

1. Which of the following is the correct syntax to declare a 3 dimensional array using pointers?

- a) char *a[][];
- b) char **a[];
- c) char ***a;

d) all of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        char *a = {"p", "r", "o", "g", "r", "a", "m"};
5.        printf("%s", a);
6.    }
```

a) Output will be program

b) Output will be p

c) No output

d) Compile-time error

[View Answer](#)

Answer: b

Explanation: None.

3. An array of strings can be initialized by _____

a) `char *a[] = {"Hello", "World"};`

b) `char *a[] = {"Hello", "Worlds"};`

c)

`char *b = "Hello";`

`char *c = "World";`

`char *a[] = {b, c};`

d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        char *a[10] = {"hi", "hello", "how"};
5.        int i = 0;
```

```
6.     for (i = 0; i < 10; i++)
7.     printf("%s", *(a[i]));
8.     }
```

- a) segmentation fault
- b) hi hello how followed by 7 null values
- c) 10 null values
- d) depends on compiler

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         char *a[10] = {"hi", "hello", "how"};
5.         int i = 0, j = 0;
6.         a[0] = "hey";
7.         for (i = 0; i < 10; i++)
8.             printf("%s\n", a[i]);
9.     }
```

- a) hi hello how Segmentation fault
- b) hi hello how followed by 7 null values
- c) hey hello how followed by 7 null values
- d) hey hello how Segmentation fault

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code on a 32-bit system?

```
1.     #include <stdio.h>
2.     void main()
3.     {
4.         char *a[10] = {"hi", "hello", "how"};
5.         printf("%d\n", sizeof(a));
6.     }
```

- a) 10
- b) 13
- c) Run time error
- d) 40

View Answer

Answer: d

Explanation: If the system is 32-bit system, then the size of pointer will be 4 bytes. For such a system, the size of array a will be $4 \times 10 = 40$ bytes. The size of pointer is 8 bytes on a 64 bit system. For the given array a of 10 elements, it will be $8 \times 10 = 80$ bytes.

7. What will be the output of the following C code on a 32-bit system?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *a[10] = {"hi", "hello", "how"};
5.      printf("%d\n", sizeof(a[1]));
6.  }
```

- a) 6
- b) 4
- c) 5
- d) 3

View Answer

Answer: b

Explanation: Array element a[1] is storing an address of character pointer. For a 32-bit systems its 4 bytes and for a 64-bit system, its 8 bytes.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *a[10] = {"hi", "hello", "how"};
5.      int i = 0;
6.      for (i = 0; i < 10; i++)
7.          printf("%s", a[i]);
8.  }
```

- a) hi hello how Segmentation fault
- b) hi hello how null
- c) hey hello how Segmentation fault

d) hi hello how followed by 7 nulls

[View Answer](#)

Answer: d

Explanation: None.

Initialization of Pointer Arrays – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        char *p[1] = {"hello"};
5.        printf("%s", (p)[0]);
6.        return 0;
7.    }
```

- a) Compile time error
- b) Undefined behaviour
- c) hello
- d) None of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        char **p = {"hello", "hi", "bye"};
5.        printf("%s", (p)[0]);
6.        return 0;
7.    }
```

- a) Compile time error
- b) Undefined behaviour
- c) hello
- d) Address of hello

[View Answer](#)

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 1;
5.      int *a[] = {&i, &j};
6.      printf("%d", (*a)[0]);
7.      return 0;
8.  }
```

- a) Compile time error
- b) Undefined behaviour
- c) 0
- d) Some garbage value

[View Answer](#)

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 1;
5.      int *a[] = {&i, &j};
6.      printf("%d", *a[0]);
7.      return 0;
8.  }
```

- a) Compile time error
- b) Undefined behaviour
- c) 0
- d) Some garbage value

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```

2.   int main()
3.   {
4.       int i = 0, j = 1;
5.       int *a[] = {&i, &j};
6.       printf("%d", (*a)[1]);
7.       return 0;
8.   }

```

- a) Compile time error
 - b) Undefined behaviour
 - c) 1
 - d) Some garbage value
- View Answer

Answer: d

Explanation: None.

6. Which of the following are generated from char pointer?

- a) char *string = "Hello.";
- b)

char *string;

scanf("%s", string);

- c) char string[] = "Hello.";
 - d) char *string = "Hello."; and char string[] = "Hello.";
- View Answer

Answer: a

Explanation: None.

7. Which of the following declaration are illegal?

- a) int a[][] = {{1, 2, 3}, {2, 3, 4, 5}};
- b) int *a[] = {{1, 2, 3}, {2, 3, 4, 5}};
- c) int a[4][4] = {{1, 2, 3}, {2, 3, 4, 5}};
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

Pointers Vs. Multi-dimensional Arrays – 1

1. Which is true for a, if a is defined as "int a[10][20];"?

- a) a is true two-dimensional array
- b) 200 int-sized locations have been set aside
- c) The conventional rectangular subscript calculation $20 * \text{row} + \text{col}$ is used to find the element a[row, col].

d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

2. Which is true for b, if b is defined as “int *b[10];”?

a) The definition only allocates 10 pointers and does not initialize them

b) Initialization must be done explicitly

c) The definition only allocates 10 pointers and does not initialize them & Initialization must be done explicitly

d) Error

[View Answer](#)

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char a[10][5] = {"hi", "hello", "fellows"};
5.      printf("%s", a[2]);
6.  }
```

a) fellows

b) fellow

c) fello

d) fell

[View Answer](#)

Answer: c

Explanation: Since every row in the array a[10][5] can contain only 5 characters, the a[2] element will hold “fello” i.e. 5 characters. There will not be any null character in a[2]. Since, the array is completely initialized, other rows (row a[3] will have only null characters. Hence, printf() using %s specifier will display fello only.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char a[10][5] = {"hi", "hello", "fellows"};
5.      printf("%p\n", a);
6.      printf("%p", a[0]);
```

7. }

- a) same address is printed
- b) different address is printed
- c) hello
- d) hi hello fello

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1. #include <stdio.h>
2. void main()
3. {
4.     char a[10][5] = {"hi", "hello", "fellows"};
5.     printf("%d", sizeof(a[1]));
6. }
```

- a) 2
- b) 4
- c) 5
- d) 10

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4.     char a[1][5] = {"hello"};
5.     printf("%s", a[0]);
6.     return 0;
7. }
```

- a) Compile time error
- b) hello
- c) Undefined behaviour
- d) hellon

View Answer

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        char *a[1] = {"hello"};
5.        printf("%s", a[0]);
6.        return 0;
7.    }
```

- a) Compile time error
- b) hello
- c) Undefined behaviour
- d) hellon

View Answer

Answer: b

Explanation: None.

8. Which of the following statements are true?

P. Pointer to Array

Q. Multi-dimensional array

- a) P are static, Q are static
- b) P are static, Q are dynamic
- c) P are dynamic, Q are static
- d) P are dynamic, Q are dynamic

View Answer

Answer: c

Explanation: None.

Pointers Vs. Multi-dimensional Arrays – 2

1. What will be the output of the following C code (considering sizeof char is 1 and pointer is 4)?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        char *a[2] = {"hello", "hi"};
5.        printf("%d", sizeof(a));
6.        return 0;
```

7. }

- a) 9
- b) 4
- c) 8
- d) 10

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4.     char a[2][6] = {"hello", "hi"};
5.     printf("%d", sizeof(a));
6.     return 0;
7. }
```

- a) 9
- b) 12
- c) 8
- d) 10

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4.     char a[2][6] = {"hello", "hi"};
5.     printf("%s", *a + 1);
6.     return 0;
7. }
```

- a) hello
- b) hi
- c) ello
- d) ello hi

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        char *a[2] = {"hello", "hi"};
5.        printf("%s", *(a + 1));
6.        return 0;
7.    }
```

- a) hello
- b) ello
- c) hi
- d) ello hi

View Answer

Answer: c

Explanation: None.

5. What is the advantage of a multidimensional array over pointer array?

- a) Predefined size
- b) Input can be taken from user
- c) Faster Access
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

6. Which of the following operation is possible using a pointer char? (Assuming the declaration is char *a;)

- a) Input via %s
- b) Generation of the multidimensional array
- c) Changing address to point at another location
- d) All of the mentioned

View Answer

Answer: c

Explanation: None.

7. Comment on the following two operations.

```
int *a[] = {{1, 2, 3}, {1, 2, 3, 4}}; //- 1
```

```
int b[4][4] = {{1, 2, 3}, {1, 2, 3, 4}}; //- 2
```

- a) 1 will work, 2 will not
- b) 1 and 2, both will work
- c) 1 won't work, 2 will work
- d) Neither of them will work

View Answer

Answer: c

Explanation: None.

8. Comment on the following two operations.

```
int *a[] = {{1, 2, 3}, {1, 2, 3, 4}}; //- 1
```

```
int b[][] = {{1, 2, 3}, {1, 2, 3, 4}}; //- 2
```

- a) 1 works, 2 doesn't
- b) 2 works, 1 doesn't
- c) Both of them work
- d) Neither of them work

View Answer

Answer: d

Explanation: None.

Command Line Arguments – 1

1. What does argc and argv indicate in command-line arguments?

(Assuming: `int main(int argc, char *argv[])`)

- a) argument count, argument variable
- b) argument count, argument vector
- c) argument control, argument variable
- d) argument control, argument vector

View Answer

Answer: b

Explanation: None.

2. Which of the following syntax is correct for command-line arguments?

a)

```
int main(int var, char *varg[])
```

b)

```
int main(char *argv[], int argc)
```

c)

```
int main()  
{  
    int argv, char *argc[];  
}
```

d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

3. In linux, argv[0] by command-line argument can be occupied by _____

a) ./a.out

b) ./test

c) ./fun.out.out

d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: All the options mentioned (./a.out, ./test, ./fun.out.out) are simply the command without any argument. A command is always stored as argument vector zero i.e., argv[0] always contain the command where as argv[1], argv[2], etc. contains the arguments to the commands, if any.

4. What type of array is generally generated in Command-line argument?

a) Single dimension array

b) 2-Dimensional Square Array

c) Jagged Array

d) 2-Dimensional Rectangular Array

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C statement? (assuming the input is "cool brother in city")

```
printf("%s\n", argv[argc]);
```

a) (null)

b) City

c) In

d) Segmentation Fault

[View Answer](#)

Answer: a

Explanation: None.

6. What is the first argument in command line arguments?

a) The number of command-line arguments the program was invoked with;

b) A pointer to an array of character strings that contain the arguments

c) Nothing

d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

7. What is the second argument in command line arguments?

- a) The number of command-line arguments the program was invoked with;
- b) A pointer to an array of character strings that contain the arguments, one per string
- c) Nothing
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

8. What is argv[0] in command line arguments?

- a) The name by which the program was invoked
- b) The name of the files which are passed to the program
- c) Count of the arguments in argv[] vector
- d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

Command Line Arguments – 2

1. A program that has no command line arguments will have argc _____

- a) Zero
- b) Negative
- c) One
- d) Two

[View Answer](#)

Answer: c

Explanation: None.

2. What is the index of the last argument in command line arguments?

- a) argc – 2
- b) argc + 1
- c) argc
- d) argc – 1

[View Answer](#)

Answer: d

Explanation: None.

3. What will be the output of the following C code (if run with no options or arguments)?

1. `#include <stdio.h>`
2. `int main(int argc, char *argv[])`
3. `{`
4. `printf("%d\n", argc);`
5. `return 0;`

6. }

- a) 0
- b) 1
- c) Depends on the platform
- d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code (run without any command line arguments)?

```
1. #include <stdio.h>
2. int main(int argc, char *argv[])
3. {
4.     while (argc--)
5.         printf("%s\n", argv[argc]);
6.     return 0;
7. }
```

- a) Compile time error
- b) Executable filename
- c) Segmentation fault
- d) Undefined

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code (run without any command line arguments)?

```
1. #include <stdio.h>
2. int main(int argc, char *argv[])
3. {
4.     printf("%s\n", argv[argc]);
5.     return 0;
6. }
```

- a) Segmentation fault/code crash
- b) Executable file name
- c) Depends on the platform
- d) Depends on the compiler

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code (run without any command line arguments)?

```
1.  #include <stdio.h>
2.  int main(int argc, char *argv[])
3.  {
4.      while (*argv++ != NULL)
5.          printf("%s\n", *argv);
6.      return 0;
7.  }
```

a) Segmentation fault/code crash

b) Executable file name

c) Depends on the platform

d) Depends on the compiler

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code (run without any command line arguments)?

```
1.  #include <stdio.h>
2.  int main(int argc, char *argv[])
3.  {
4.      while (*argv != NULL)
5.          printf("%s\n", *(argv++));
6.      return 0;
7.  }
```

a) Segmentation fault/code crash

b) Executable file name

c) Depends on the platform

d) Depends on the compiler

[View Answer](#)

Answer: b

Explanation: None.

8. What will be the output of the following C code (run without any command line arguments)?

```
1.  #include <stdio.h>
2.  int main(int argc, char *argv[])
```



```
3.  {
4.      while (argv != NULL)
5.          printf("%s\n", *(argv++));
6.      return 0;
7.  }
```

- a) Segmentation fault/code crash
- b) Executable file name
- c) Depends on the platform
- d) Depends on the compiler

View Answer

Answer: a

Explanation: None.

Pointers to Functions – 1

1. Which function is not called in the following C program?

```
1.  #include <stdio.h>
2.  void first()
3.  {
4.      printf("first");
5.  }
6.  void second()
7.  {
8.      first();
9.  }
10. void third()
11. {
12.     second();
13. }
14. void main()
15. {
16.     void (*ptr)();
17.     ptr = third;
18.     ptr();
19. }
```

- a) Function first
- b) Function second
- c) Function third
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

2. How to call a function without using the function name to send parameters?

- a) typedefs
- b) Function pointer
- c) Both typedefs and Function pointer
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

3. Which of the following is a correct syntax to pass a Function Pointer as an argument?

- a) void pass(int (*fptr)(int, float, char)){}
- b) void pass(*fptr(int, float, char)){}
- c) void pass(int (*fptr)){}
- d) void pass(*fptr){}

View Answer

Answer: a

Explanation: None.

4. Which of the following is not possible in C?

- a) Array of function pointer
- b) Returning a function pointer
- c) Comparison of function pointer
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

1. `#include <stdio.h>`
2. `void first()`
3. `{`
4. `printf("Hello World");`
5. `}`
6. `void main()`
7. `{`

```

8.    void *ptr() = first;
9.    ptr++
10.   ptr();
11.   }

```

- a) Illegal application of ++ to void data type
- b) pointer function initialized like a variable
- c) Illegal application of ++ to void data type & pointer function initialized like a variable
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int mul(int a, int b, int c)
3.    {
4.        return a * b * c;
5.    }
6.    void main()
7.    {
8.        int (*function_pointer)(int, int, int);
9.        function_pointer = mul;
10.       printf("The product of three numbers is:%d",
11.           function_pointer(2, 3, 4));
12.    }

```

- a) The product of three numbers is:24
- b) Run time error
- c) Nothing
- d) Varies

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int mul(int a, int b, int c)

```

```

3.  {
4.      return a * b * c;
5.  }
6.  void main()
7.  {
8.      int (function_pointer)(int, int, int);
9.      function_pointer = mul;
10.     printf("The product of three numbers is:%d",
11.         function_pointer(2, 3, 4));
12. }

```

- a) The product of three numbers is:24
- b) Compile time error
- c) Nothing
- d) Varies

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void f(int (*x)(int));
3.  int myfoo(int);
4.  int (*fooptr)(int);
5.  int ((*foo(int)))(int);
6.  int main()
7.  {
8.      fooptr = foo(0);
9.      fooptr(10);
10. }
11. int ((*foo(int i)))(int)
12. {
13.     return myfoo;
14. }
15. int myfoo(int i)

```

```
16.  {  
17.    printf("%d\n", i + 1);  
18.  }
```

- a) 10
- b) 11
- c) Compile time error
- d) Undefined behaviour

[View Answer](#)

Answer: b

Explanation: None.

Pointers to Functions – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>  
2.  int mul(int a, int b, int c)  
3.  {  
4.      return a * b * c;  
5.  }  
6.  void main()  
7.  {  
8.      int *function_pointer;  
9.      function_pointer = mul;  
10.     printf("The product of three numbers is:%d",  
11.         function_pointer(2, 3, 4));  
12. }
```

- a) The product of three numbers is:24
- b) Compile time error
- c) Nothing
- d) Varies

[View Answer](#)

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>  
2.  int sub(int a, int b, int c)
```

```

3.  {
4.      return a - b - c;
5.  }
6.  void main()
7.  {
8.      int (*function_pointer)(int, int, int);
9.      function_pointer = &sub;
10.     printf("The difference of three numbers is:%d",
11.         (*function_pointer)(2, 3, 4));
12. }

```

- a) The difference of three numbers is:1
- b) Run time error
- c) The difference of three numbers is:-5
- d) Varies

View Answer

Answer: c

Explanation: None.

3. One of the uses for function pointers in C is _____

- a) Nothing
- b) There are no function pointers in c
- c) To invoke a function
- d) To call a function defined at run-time

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void f(int);
3.  void (*foo)() = f;
4.  int main(int argc, char *argv[])
5.  {
6.      foo(10);
7.      return 0;
8.  }
9.  void f(int i)

```

```
10.  {  
11.    printf("%d\n", i);  
12.  }
```

- a) Compile time error
- b) 10
- c) Undefined behaviour
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>  
2.  void f(int);  
3.  void (*foo)(void) = f;  
4.  int main(int argc, char *argv[])  
5.  {  
6.    foo(10);  
7.    return 0;  
8.  }  
9.  void f(int i)  
10. {  
11.    printf("%d\n", i);  
12. }
```

- a) Compile time error
- b) 10
- c) Undefined behaviour
- d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>  
2.  void f(int);  
3.  void (*foo)(float) = f;
```

```

4.   int main()
5.   {
6.       foo(10);
7.   }
8.   void f(int i)
9.   {
10.      printf("%d\n", i);
11.  }

```

- a) Compile time error
 - b) 10
 - c) 10.000000
 - d) Undefined behaviour
- View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   void f(int (*x)(int));
3.   int myfoo(int i);
4.   int (*foo)(int) = myfoo;
5.   int main()
6.   {
7.       f(foo(10));
8.   }
9.   void f(int (*i)(int))
10.  {
11.      i(11);
12.  }
13.  int myfoo(int i)
14.  {
15.      printf("%d\n", i);
16.      return i;
17.  }

```


- a) Compile time error
- b) Undefined behaviour
- c) 10 11
- d) 10 Segmentation fault

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void f(int (*x)(int));
3.  int myfoo(int);
4.  int (*foo)() = myfoo;
5.  int main()
6.  {
7.      f(foo);
8.  }
9.  void f(int(*i)(int ))
10. {
11.     i(11);
12. }
13. int myfoo(int i)
14. {
15.     printf("%d\n", i);
16.     return i;
17. }
```

- a) 10 11
- b) 11
- c) 10
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

Complicated Declarations – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      struct student
5.      {
6.          int no;
7.          char name[20];
8.      };
9.      struct student s;
10.     no = 8;
11.     printf("%d", no);
12. }
```

- a) Nothing
- b) Compile time error
- c) Junk
- d) 8

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      int no;
5.      char name[20];
6.  };
7.  void main()
8.  {
9.      struct student s;
10.     s.no = 8;
11.     printf("hello");
12. }
```

- a) Run time error
- b) Nothing
- c) hello
- d) Varies

View Answer

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct student
4.  {
5.      int no = 5;
6.      char name[20];
7.  };
8.  void main()
9.  {
10.     struct student s;
11.     s.no = 8;
12.     printf("hello");
13. }
```

- a) Nothing
- b) Compile time error
- c) hello
- d) Varies

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct student
4.  {
5.      int no;
6.      char name[20];
7.  };
8.  void main()
9.  {
10.     struct student s;
11.     s.no = 8;
12.     printf("hello");
13. }
```

```
7. void main()
8. {
9.     student s;
10.    s.name = "hello";
11.    printf("hello");
12. }
```

- a) Nothing
- b) hello
- c) Compile time error
- d) Varies

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1. #include <stdio.h>
2. void main()
3. {
4.     struct student
5.     {
6.         int no;
7.         char name[20];
8.     };
9.     struct student s;
10.    s.no = 8;
11.    printf("%s", s.name);
12. }
```

- a) Nothing
- b) Compile time error
- c) Junk
- d) 8

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct student
3.  {
4.      int no;
5.      char name[20];
6.  };
7.  struct student s;
8.  void main()
9.  {
10.     s.no = 8;
11.     printf("%s", s.name);
12. }

```

- a) Nothing
- b) Compile time error
- c) Junk
- d) 8

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int *((*x)())[2];
5.      x();
6.      printf("after x\n");
7.  }
8.  int *((*x)())[2]
9.  {
10.     int **str;
11.     str = (int*)malloc(sizeof(int)* 2);
12.     return str;
13. }

```

- a) Compile time error
- b) Undefined behaviour
- c) After x
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void (*(f)))(int, float);
3.  void (*(x)))(int, float) = f;
4.  void ((*y)(int, float));
5.  void foo(int i, float f);
6.  int main()
7.  {
8.      y = x();
9.      y(1, 2);
10. }
11. void (*(f)))(int, float)
12. {
13.     return foo;
14. }
15. void foo(int i, float f)
16. {
17.     printf("%d %f\n", i, f);
18. }
```

- a) 1 2.000000
- b) 1 2
- c) Compile time error
- d) Segmentation fault/code crash

View Answer

Answer: a

Explanation: None.

9. What does this declaration say?

```
int (*(*y)())[2];
```

- a) y is pointer to the function which returns pointer to integer array
- b) y is pointer to the function which returns array of pointers
- c) y is function which returns function pointer which in turn returns pointer to integer array
- d) y is function which returns array of integers

View Answer

Answer: a

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void (*f())(int, float);
3.  typedef void (*x())(int, float);
4.  void foo(int i, float f);
5.  int main()
6.  {
7.      x = f;
8.      x();
9.  }
10. void (*f())(int, float)
11. {
12.     return foo;
13. }
14. void foo(int i, float f)
15. {
16.     printf("%d %f\n", i, f);
17. }
```

- a) Compile time error
- b) Undefined behaviour
- c) 1 2.000000
- d) Nothing

View Answer

Answer: a

Explanation: None.

11. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void (*(f)) (int, float);
3.  typedef void (*(x)) (int, float);
4.  void foo(int i, float f);
5.  int main()
6.  {
7.      x p = f;
8.      p();
9.  }
10. void (*(f)) (int, float)
11. {
12.     return foo;
13. }
14. void foo(int i, float f)
15. {
16.     printf("%d %f\n", i, f);
17. }

```

- a) Compile time error
- b) Undefined behaviour
- c) 1 2.000000
- d) Nothing

View Answer

Answer: d

Explanation: None.

Complicated Declarations – 2

1. Read the following expression?

```
void (*ptr)(int);
```

- a) ptr is pointer to int that converts its type to void
- b) ptr is pointer to function passing int returning void
- c) ptr is pointer to void that converts its type to int
- d) ptr is pointer to function passing void returning int

View Answer

Answer: b

Explanation: None.

2. Which of the following expression is true for the following C statement?

ptr is array with 3 elements of pointer to function returning pointer of int

- a) `int **ptr[3]();`
- b) `int *(*ptr[3])();`
- c) `int (*(*ptr[3]))();`
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

3. What makes the following declaration denote?

`int **ptr;`

- a) ptr is a function pointer that returns pointer to int type
- b) ptr is a pointer to an int pointer
- c) ptr is a pointer to pointer to type int
- d) none of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

4. What makes the following declaration denote?

`char *str[5];`

- a) str is an array of 5 element pointer to type char
- b) str is a pointer to an array of 5 elements
- c) str is a function pointer of 5 elements returning char
- d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

5. Comment on the following declaration.

`int (*ptr)(); // i)`

`char *ptr[]; // ii)`

- a) Both i) and ii) and cannot exist due to same name
- b) i) is legal, ii) is illegal
- c) i) is illegal, ii) is legal
- d) Both i) and ii) will work legal and flawlessly

[View Answer](#)

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      int no;
5.      char name[20];
6.  }
7.  void main()
8.  {
9.      struct student s;
10.     s.no = 8;
11.     printf("hello");
12. }
```

- a) Compile time error
- b) Nothing
- c) hello
- d) Varies

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      int no = 5;
5.      char name[20];
6.  };
7.  void main()
8.  {
9.      struct student s;
10.     s.no = 8;
11.     printf("hello");
12. }
```

- a) Nothing
- b) Compile time error
- c) hello
- d) Varies

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct student
4.  {
5.      int no;
6.      char name[20];
7.  };
8.  void main()
9.  {
10.     student s;
11.     s.no = 8;
12.     printf("hello");
13. }
```

- a) Nothing
- b) hello
- c) Compile time error
- d) Varies

View Answer

Answer: c

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  void main()
4.  {
5.     struct student
6.     {
7.         int no;
```

```
7.      char name[20];  
8.      };  
9.      struct student s;  
10.     s.no = 8;  
11.     printf("%d", s.no);  
12. }
```

- a) Nothing
- b) Compile time error
- c) Junk
- d) 8

View Answer

Answer: d

Explanation: None.

10. Is the below declaration legal?

```
int* ((*x())[2]);
```

- a) True
- b) False
- c) Undefined behaviour
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

C MCQs on Structures, Unions and Bit-Fields

Basics of Structures – 1

1. Which of the following are themselves a collection of different data types?

- a) string
- b) structures
- c) char
- d) all of the mentioned

View Answer

Answer: b

Explanation: None.

2. User-defined data type can be derived by_____

- a) struct
- b) enum
- c) typedef
- d) all of the mentioned

View Answer

Answer: d

Explanation: None.

3. Which operator connects the structure name to its member name?

- a) –
- b) <-
- c) .
- d) Both <- and .

View Answer

Answer: c

Explanation: None.

4. Which of the following cannot be a structure member?

- a) Another structure
- b) Function
- c) Array
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

5. Which of the following structure declaration will throw an error?

a)

```
struct temp{ }s;
```

```
main(){ }
```

b)

```
struct temp{};

struct temp s;

main(){}
```

c)

```
struct temp s;

struct temp{};

main(){}
```

d) None of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>

2.  struct student

3.  {

4.      int no;

5.      char name[20];

6.  }

7.  void main()

8.  {

9.      struct student s;

10.     s.no = 8;

11.     printf("hello");

12. }
```

a) Compile time error

b) Nothing

c) hello

d) Varies

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```
2.    struct student
3.    {
4.        int no = 5;
5.        char name[20];
6.    };
7.    void main()
8.    {
9.        struct student s;
10.       s.no = 8;
11.       printf("hello");
12.    }
```

- a) Nothing
 - b) Compile time error
 - c) hello
 - d) Varies
- [View Answer](#)

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    struct student
3.    {
4.        int no;
5.        char name[20];
6.    };
7.    void main()
8.    {
9.        student s;
10.       s.no = 8;
11.       printf("hello");
12.    }
```

- a) Nothing
- b) hello

c) Compile time error

d) Varies

View Answer

Answer: c

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      struct student
5.      {
6.          int no;
7.          char name[20];
8.      };
9.      struct student s;
10.     s.no = 8;
11.     printf("%d", s.no);
12. }
```

a) Nothing

b) Compile time error

c) Junk

d) 8

View Answer

Answer: d

Explanation: None.

10. Can the following C code be compiled successfully?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int k;
5.      char c;
6.      float f;
7.  };
8.  int main()
```



```

9.    {
10.    struct p x = {.c = 97, .f = 3, .k = 1};
11.    printf("%f\n", x.f);
12.    }

```

- a) Yes
- b) No
- c) Depends on the standard
- d) Depends on the platform

View Answer

Answer: c

Explanation: None.

Basics of Structures – 2

1. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.
3.    void main()
4.    {
5.        struct student
6.        {
7.            int no;
8.            char name[20];
9.        };
10.     struct student s;
11.     no = 8;
12.     printf("%d", no);

```

- a) Nothing
- b) Compile time error
- c) Junk
- d) 8

View Answer

Answer: b

Explanation: None.

2. How many bytes in memory taken by the following C structure?

```

1.    #include <stdio.h>

```

```

2.    struct test
3.    {
4.        int k;
5.        char c;
6.    };

```

- a) Multiple of integer size
- b) integer size+character size
- c) Depends on the platform
- d) Multiple of word size

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    struct
3.    {
4.        int k;
5.        char c;
6.    };
7.    int main()
8.    {
9.        struct p;
10.       p.k = 10;
11.       printf("%d\n", p.k);
12.    }

```

- a) Compile time error
- b) 10
- c) Undefined behaviour
- d) Segmentation fault

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```

1.    #include <stdio.h>

```

```

2.    struct
3.    {
4.        int k;
5.        char c;
6.    } p;
7.    int p = 10;
8.    int main()
9.    {
10.        p.k = 10;
11.        printf("%d %d\n", p.k, p);
12.    }

```

- a) Compile time error
 - b) 10 10
 - c) Depends on the standard
 - d) Depends on the compiler
- [View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    struct p
3.    {
4.        int k;
5.        char c;
6.    };
7.    int p = 10;
8.    int main()
9.    {
10.        struct p x;
11.        x.k = 10;
12.        printf("%d %d\n", x.k, p);
13.    }

```

- a) Compile time error
- b) 10 10
- c) Depends on the standard
- d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct p
4.  {
5.      int k;
6.      char c;
7.      float f;
8.  };
9.  int p = 10;
10. int main()
11. {
12.     struct p x = {1, 97};
13.     printf("%f %d\n", x.f, p);
14. }
```

- a) Compile time error
- b) 0.000000 10
- c) Somegarbage value 10
- d) 0 10

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code according to C99 standard?

```
1.  #include <stdio.h>
2.
3.  struct p
4.  {
5.      int k;
```

```

6.    float f;
7.    };
8.    int main()
9.    {
10.    struct p x = {.c = 97, .f = 3, .k = 1};
11.    printf("%f\n", x.f);
12.    }

```

- a) 3.000000
 - b) Compile time error
 - c) Undefined behaviour
 - d) 1.000000
- [View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code according to C99 standard?

```

1.    #include <stdio.h>
2.    struct p
3.    {
4.    int k;
5.    char c;
6.    float f;
7.    };
8.    int main()
9.    {
10.    struct p x = {.c = 97, .k = 1, 3};
11.    printf("%f \n", x.f);
12.    }

```

- a) 3.000000
 - b) 0.000000
 - c) Compile time error
 - d) Undefined behaviour
- [View Answer](#)

Answer: b

Explanation: None.

9. What will be the output of the following C code according to C99 standard?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int k;
5.      char c;
6.      float f;
7.  };
8.  int main()
9.  {
10.     struct p x = {.c = 97};
11.     printf("%f\n", x.f);
12. }
```

- a) 0.000000
- b) Somegarbagevalue
- c) Compile time error
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

Structures and Functions – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  struct student s;
7.  struct student fun(void)
8.  {
9.      s.name = "newton";
10.     printf("%s\n", s.name);
11.     s.name = "alan";
```

```

12.     return s;
13. }
14. void main()
15. {
16.     struct student m = fun();
17.     printf("%s\n", m.name);
18.     m.name = "turing";
19.     printf("%s\n", s.name);
20. }

```

- a) newton alan alan
- b) alan newton alan
- c) alan alan newton
- d) compile time error

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  void main()
7.  {
8.      struct student s, m;
9.      s.name = "st";
10.     m = s;
11.     printf("%s%s", s.name, m.name);
12. }

```

- a) Compile time error
- b) Nothing
- c) Junk values
- d) st st

View Answer

Answer: d

Explanation: None.

3. Which of the following return-type cannot be used for a function in C?

- a) char *
- b) struct
- c) void
- d) none of the mentioned

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct temp
4.  {
5.      int a;
6.  } s;
7.  void func(struct temp s)
8.  {
9.      s.a = 10;
10.     printf("%d\t", s.a);
11. }
12. main()
13. {
14.     func(s);
15.     printf("%d\t", s.a);
16. }
```

- a) 10 (Garbage Value)
- b) 0 10
- c) 10 0
- d) (Garbage Value) 10

View Answer

Answer: c

Explanation: None.

5. Which of the following is not possible under any scenario?

- a) s1 = &s2;
- b) s1 = s2;

c) (*s1).number = 10;
d) None of the mentioned
View Answer

Answer: d
Explanation: None.

6. Which of the following operation is illegal in structures?
a) Typecasting of structure
b) Pointer to a variable of the same structure
c) Dynamic allocation of memory for structure
d) All of the mentioned
View Answer

Answer: a
Explanation: None.

7. Presence of code like "s.t.b = 10" indicates _____
a) Syntax Error
b) Structure
c) double data type
d) An ordinary variable name
View Answer

Answer: b
Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct student
4.  {
5.      char *name;
6.  };
7.  struct student fun(void)
8.  {
9.      struct student s;
10.     s.name = "alan";
11.     return s;
12. }
13. void main()
14. {
15.     struct student m = fun();
```

```
15.     s.name = "turing";
16.     printf("%s", m.name);
17. }
```

- a) alan
- b) Turing
- c) Compile time error
- d) Nothing

[View Answer](#)

Answer: c

Explanation: None.

Structures and Functions – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct point
4.  {
5.      int x;
6.      int y;
7.  };
8.  int main()
9.  {
10.     struct point p = {1};
11.     struct point p1 = {1};
12.     if(p == p1)
13.         printf("equal\n");
14.     else
15.         printf("not equal\n");
16. }
```

- a) Compile time error
- b) equal
- c) depends on the standard
- d) not equal

[View Answer](#)

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  struct notpoint
8.  {
9.      int x;
10.     int y;
11. };
12. struct point foo();
13. int main()
14. {
15.     struct point p = {1};
16.     struct notpoint p1 = {2, 3};
17.     p1 = foo();
18.     printf("%d\n", p1.x);
19. }
20. struct point foo()
21. {
22.     struct point temp = {1, 2};
23.     return temp;
24. }
```

- a) Compile time error
- b) 1
- c) 2
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  struct notpoint
8.  {
9.      int x;
10.     int y;
11. };
12. int main()
13. {
14.     struct point p = {1};
15.     struct notpoint p1 = p;
16.     printf("%d\n", p1.x);
17. }

```

a) Compile time error

b) 1

c) 0

d) Undefined

[View Answer](#)

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  struct notpoint
8.  {

```

```

9.     int x;
10.    int y;
11.    };
12.    void foo(struct point);
13.    int main()
14.    {
15.        struct notpoint p1 = {1, 2};
16.        foo(p1);
17.    }
18.    void foo(struct point p)
19.    {
20.        printf("%d\n", p.x);
21.    }

```

a) Compile time error

b) 1

c) 0

d) Undefined

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    struct point
3.    {
4.        int x;
5.        int y;
6.    };
7.    void foo(struct point*);
8.    int main()
9.    {
10.        struct point p1 = {1, 2};
11.        foo(&p1);
12.    }

```

```

13. void foo(struct point *p)
14. {
15.     printf("%d\n", *p.x++);
16. }

```

- a) Compile time error
- b) Segmentation fault/code crash
- c) 2
- d) 1

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.
3.  struct point
4.  {
5.      int x;
6.      int y;
7.  };
8.  void foo(struct point*);
9.  int main()
10. {
11.     struct point p1 = {1, 2};
12.     foo(&p1);
13. }
14. void foo(struct point *p)
15. {
16.     printf("%d\n", *p->x++);
17. }

```

- a) Compile time error
- b) 1
- c) Segmentation fault/code crash
- d) 2

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student fun(void)
3.  {
4.      struct student
5.      {
6.          char *name;
7.      };
8.      struct student s;
9.      s.name = "alan";
10.     return s;
11. }
12. void main()
13. {
14.     struct student m = fun();
15.     printf("%s", m.name);
16. }
```

a) Compile time error

b) alan

c) Nothing

d) Varies

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  struct student fun(void)
```

```

7.  {
8.      struct student s;
9.      s.name = "alan";
10.     return s;
11. }
12. void main()
13. {
14.     struct student m = fun();
15.     printf("%s", m.name);
16. }

```

- a) Nothing
- b) alan
- c) Run time error
- d) Varies

View Answer

Answer: b

Explanation: None.

Arrays of Structures – 1

1. The correct syntax to access the member of the ith structure in the array of structures is?

Assuming: struct temp

```

{
    int b;
}s[50];

```

- a) s.b.[i];
- b) s.[i].b;
- c) s.b[i];
- d) s[i].b;

View Answer

Answer: d

Explanation: None.

2. Comment on the output of the following C code.

```

1.  #include <stdio.h>
2.  struct temp
3.  {

```



```

4.     int a;
5.     int b;
6.     int c;
7.     };
8.     main()
9.     {
10.    struct temp p[] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
11.    }

```

- a) No Compile time error, generates an array of structure of size 3
- b) No Compile time error, generates an array of structure of size 9
- c) Compile time error, illegal declaration of a multidimensional array
- d) Compile time error, illegal assignment to members of structure

View Answer

Answer: a

Explanation: None.

3. Which of the following uses structure?

- a) Array of structures
- b) Linked Lists
- c) Binary Tree
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

4. What is the correct syntax to declare a function foo() which receives an array of structure in function?

- a) void foo(struct *var);
- b) void foo(struct *var[]);
- c) void foo(struct var);
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code? (Assuming size of int be 4)

```

1.    #include <stdio.h>
2.
3.    struct temp
4.    {
5.        int a;

```

```

5.     int b;
6.     int c;
7.     } p[] = {0};
8.     main()
9.     {
10.    printf("%d", sizeof(p));
11.    }

```

- a) 4
 - b) 12
 - c) 16
 - d) Can't be estimated due to ambiguous initialization of array
- [View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     struct student
3.     {
4.         char *name;
5.     };
6.     struct student s[2];
7.     void main()
8.     {
9.         s[0].name = "alan";
10.        s[1] = s[0];
11.        printf("%s%s", s[0].name, s[1].name);
12.        s[1].name = "turing";
13.        printf("%s%s", s[0].name, s[1].name);
14.    }

```

- a) alan alan alan turing
- b) alan alan turing turing
- c) alan turing alan turing
- d) run time error

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  struct student s[2], r[2];
7.  void main()
8.  {
9.      s[0].name = "alan";
10.     s[1] = s[0];
11.     r = s;
12.     printf("%s%s", r[0].name, r[1].name);
13. }
```

- a) alan alan
- b) Compile time error
- c) Varies
- d) Nothing

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  void main()
7.  {
8.      struct student s[2], r[2];
9.      s[1] = s[0] = "alan";
```

10. printf("%s%s", s[0].name, s[1].name);

11. }

- a) alan alan
- b) Nothing
- c) Compile time error
- d) Varies

View Answer

Answer: c

Explanation: None.

9. What will be the output of the following C code?

```
1.   #include <stdio.h>
2.   struct student
3.   {
4.   };
5.   void main()
6.   {
7.       struct student s[2];
8.       printf("%d", sizeof(s));
9.   }
```

- a) 2
- b) 4
- c) 8
- d) 0

View Answer

Answer: d

Explanation: None.

Arrays of Structures – 2

1. What will be the output of the following C code?

```
1.   #include <stdio.h>
2.   struct point
3.   {
4.       int x;
5.       int y;
6.   };
```

```

7. void foo(struct point*);
8. int main()
9. {
10.     struct point p1[] = {1, 2, 3, 4};
11.     foo(p1);
12. }
13. void foo(struct point p[])
14. {
15.     printf("%d\n", p[1].x);
16. }

```

- a) Compile time error
- b) 3
- c) 2
- d) 1

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```

1. #include <stdio.h>
2. struct point
3. {
4.     int x;
5.     int y;
6. };
7. void foo(struct point*);
8. int main()
9. {
10.     struct point p1[] = {1, 2, 3, 4};
11.     foo(p1);
12. }
13. void foo(struct point p[])
14. {
15.     printf("%d\n", p->x);

```

16. }

- a) 1
- b) 2
- c) 3
- d) Compile time error

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  struct point
4.  {
5.      int x;
6.      int y;
7.  };
8.  void foo(struct point*);
9.  int main()
10. {
11.     struct point p1[] = {1, 2, 3, 4};
12.     foo(p1);
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d %d\n", p->x, ++p->x);
```

- a) 1 2
- b) 2 2
- c) Compile time error
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```

2.    struct point
3.    {
4.        int x;
5.        int y;
6.    } p[] = {1, 2, 3, 4, 5};
7.    void foo(struct point*);
8.    int main()
9.    {
10.        foo(p);
11.    }
12.    void foo(struct point p[])
13.    {
14.        printf("%d %d\n", p->x, p[2].y);
15.    }

```

- a) 1 0
- b) Compile time error
- c) 1 somegarbagevalue
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    struct point
3.    {
4.        int x;
5.        int y;
6.    };
7.    void foo(struct point*);
8.    int main()
9.    {
10.        struct point p1[] = {1, 2, 3, 4, 5};
11.        foo(p1);

```

```

12. }
13. void foo(struct point p[])
14. {
15.     printf("%d %d\n", p->x, p[3].y);
16. }

```

- a) Compile time error
- b) 1 0
- c) 1 somegarbagevalue
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4, 5};
11.     foo(p1);
12. }
13. void foo(struct point p[])
14. {
15.     printf("%d %d\n", p->x, (p + 2).y);
16. }

```

- a) Compile time error
- b) 1 0
- c) 1 somegarbagevalue
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4, 5};
11.     foo(p1);
12. }
13. void foo(struct point p[])
14. {
15.     printf("%d %d\n", p->x, (p + 2)->y);
16. }
```

a) Compile time error

b) 1 0

c) 1 somegarbagevalue

d) undefined behaviour

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code on a 64-bit system?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.  };
6.  void main()
```

```

7.  {
8.      struct student s[2];
9.      printf("%d", sizeof(s));
10. }

```

- a) 2
- b) 4
- c) 16
- d) 8

View Answer

Answer: c

Explanation: On a 64-bit system, size of pointer is 8 bytes. Here, we are printing the size of an array of 2 structures, hence, the size will be $2 \times 8 = 16$ bytes.

Pointer to Structures – 1

1. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.
3.  struct p
4.  {
5.      int x;
6.      char y;
7.  };
8.  int main()
9.  {
10.     struct p p1[] = {1, 92, 3, 94, 5, 96};
11.     struct p *ptr1 = p1;
12.     int x = (sizeof(p1) / 3);
13.     if (x == sizeof(int) + sizeof(char))
14.         printf("%d\n", ptr1->x);
15.     else
16.         printf("falsen");

```

- a) Compile time error
- b) 1
- c) Undefined behaviour

d) false

[View Answer](#)

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
5.      char y;
6.  };
7.  int main()
8.  {
9.      struct p p1[] = {1, 92, 3, 94, 5, 96};
10.     struct p *ptr1 = p1;
11.     int x = (sizeof(p1) / sizeof(ptr1));
12.     if (x == 1)
13.         printf("%d\n", ptr1->x);
14.     else
15.         printf("false\n");
16. }
```

a) Compile time error

b) 1

c) false

d) Undefined behaviour

[View Answer](#)

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
```

```

5.     char y;
6. };
7.     typedef struct p* q*;
8.     int main()
9.     {
10.        struct p p1[] = {1, 92, 3, 94, 5, 96};
11.        q ptr1 = p1;
12.        printf("%d\n", ptr1->x);
13.    }

```

- a) Compile time error
 - b) 1
 - c) Undefined behaviour
 - d) Segmentation fault
- View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     struct p
3.     {
4.         int x;
5.         char y;
6.     };
7.     void foo(struct p* );
8.     int main()
9.     {
10.        typedef struct p* q;
11.        struct p p1[] = {1, 92, 3, 94, 5, 96};
12.        foo(p1);
13.    }
14.    void foo(struct p* p1)
15.    {
16.        q ptr1 = p1;

```

17. `printf("%d\n", ptr1->x);`

18. `}`

- a) Compile time error
- b) 1
- c) Segmentation fault
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

5. Which of the following is an incorrect syntax for pointer to structure?

(Assuming `struct temp{int b;}*my_struct;`)

- a) `*my_struct.b = 10;`
- b) `(*my_struct).b = 10;`
- c) `my_struct->b = 10;`
- d) Both `*my_struct.b = 10;` and `(*my_struct).b = 10;`

View Answer

Answer: a

Explanation: None.

6. Which of the following is an incorrect syntax to pass by reference a member of a structure in a function?

(Assume: `struct temp{int a;}s;`)

- a) `func(&s.a);`
- b) `func(&(s).a);`
- c) `func(&(s.a));`
- d) none of the mentioned

View Answer

Answer: d

Explanation: None.

7. Which of the following structure declaration doesn't require pass-by-reference?

a)

```
struct{int a;}s;
```

```
main(){}
```

b)

```
struct temp{int a;;}
```

```
main(){
```

```
    struct temp s;
```

```
}
```

c)

```
struct temp{int a;};
```

```
main(){}
```

```
struct temp s;
```

d) none of the mentioned

View Answer

Answer: d

Explanation: None.

8. Which option is not possible for the following function call?

1. `func(&s.a);` //where *s* is a variable of type struct and *a* is the member of the struct.

a) Compiler can access entire structure from the function

b) Individual member's address can be displayed in structure

c) Individual member can be passed by reference in a function

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

9. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `struct temp`

3. `{`

4. `int a;`

5. `} s;`

6. `void change(struct temp);`

7. `main()`

8. `{`

9. `s.a = 10;`

10. `change(s);`

11. `printf("%d\n", s.a);`

12. `}`

13. `void change(struct temp s)`

14. `{`

15. `s.a = 1;`

16. }

- a) Output will be 1
- b) Output will be 10
- c) Output varies with machine
- d) Compile time error

View Answer

Answer: b

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
5.      int y;
6.  };
7.  int main()
8.  {
9.      struct p p1[] = {1, 92, 3, 94, 5, 96};
10.     struct p *ptr1 = p1;
11.     int x = (sizeof(p1) / 5);
12.     if (x == 3)
13.         printf("%d %d\n", ptr1->x, (ptr1 + x - 1)->x);
14.     else
15.         printf("false\n");
16. }
```

- a) Compile time error
- b) 1 5
- c) Undefined behaviour
- d) false

View Answer

Answer: d

Explanation: None.

Pointer to Structures – 2

1. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.  };
6.  void main()
7.  {
8.      struct student m;
9.      struct student *s = &m;
10.     s->c = "hello";
11.     printf("%s", s->c);
12. }

```

- a) hello
- b) Run time error
- c) Nothing
- d) Depends on compiler

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.  };
6.  void main()
7.  {
8.      struct student *s;
9.      s->c = "hello";
10.     printf("%s", s->c);
11. }

```

- a) hello
- b) Segmentation fault

c) Run time error

d) Nothing

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.  };
6.  void main()
7.  {
8.      struct student m;
9.      struct student *s = &m;
10.     s->c = "hello";
11.     printf("%s", m.c);
12. }
```

a) Run time error

b) Nothing

c) hello

d) Varies

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.  };
6.  void main()
7.  {
8.      struct student m;
```

```

9.     struct student *s = &m;
10.    (*s).c = "hello";
11.    printf("%s", m.c);
12.    }

```

- a) Run time error
- b) Nothing
- c) Varies
- d) hello

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.
3.    struct student
4.    {
5.        char *c;
6.    };
7.
8.    void main()
9.    {
10.        struct student n;
11.        struct student *s = &n;
12.        (*s).c = "hello";
13.        printf("%p\n%p\n", s, &n);
14.    }

```

- a) Different address
- b) Run time error
- c) Nothing
- d) Same address

View Answer

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.
3.    struct p

```

```

3.  {
4.      int x[2];
5.  };
6.  struct q
7.  {
8.      int *x;
9.  };
10. int main()
11. {
12.     struct p p1 = {1, 2};
13.     struct q *ptr1;
14.     ptr1->x = (struct q*)&p1.x;
15.     printf("%d\n", ptr1->x[1]);
16. }

```

- a) Compile time error
- b) Segmentation fault/code crash
- c) 2
- d) 1

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x[2];
5.  };
6.  struct q
7.  {
8.      int *x;
9.  };
10. int main()
11. {

```

```

12.     struct p p1 = {1, 2};
13.     struct q *ptr1 = (struct q*)&p1;
14.     ptr1->x = (struct q*)&p1.x;
15.     printf("%d\n", ptr1->x[0]);
16. }

```

- a) Compile time error
- b) Undefined behaviour
- c) Segmentation fault/code crash
- d) 1

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     struct p
3.     {
4.         int x;
5.         int y;
6.     };
7.     int main()
8.     {
9.         struct p p1[] = {1, 2, 3, 4, 5, 6};
10.        struct p *ptr1 = p1;
11.        printf("%d %d\n", ptr1->x, (ptr1 + 2)->x);
12.    }

```

- a) 1 5
- b) 1 3
- c) Compile time error
- d) 1 4

View Answer

Answer: a

Explanation: None.

9. What will be the output of the following C code?

```

1.     #include <stdio.h>

```

```

2.    struct p
3.    {
4.        int x;
5.        char y;
6.    };
7.    int main()
8.    {
9.        struct p p1[] = {1, 92, 3, 94, 5, 96};
10.       struct p *ptr1 = p1;
11.       int x = (sizeof(p1) / sizeof(struct p));
12.       printf("%d %d\n", ptr1->x, (ptr1 + x - 1)->x);
13.    }

```

- a) Compile time error
- b) Undefined behaviour
- c) 1 3
- d) 1 5

View Answer

Answer: d

Explanation: None.

Self-Referential Structures – 1

1. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    struct student
3.    {
4.        char *c;
5.        struct student *point;
6.    };
7.    void main()
8.    {
9.        struct student s;
10.       struct student m;
11.       s.c = m.c = "hi";
12.       m.point = &s;

```

```

13.     (m.point)->c = "hey";
14.     printf("%s\t%s\t", s.c, m.c);
15. }

```

- a) hey hi
- b) hi hey
- c) Run time error
- d) hey hey

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.
3.     struct student
4.     {
5.         char *c;
6.         struct student *point;
7.     };
8.     void main()
9.     {
10.        struct student s;
11.        struct student m;
12.        m.point = s;
13.        (m.point)->c = "hey";
14.        printf("%s", s.c);
15.    }

```

- a) Nothing
- b) Compile time error
- c) hey
- d) Varies

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.     #include <stdio.h>

```

```
2.    struct student
3.    {
4.        char *c;
5.        struct student point;
6.    };
7.    void main()
8.    {
9.        struct student s;
10.        s.c = "hello";
11.        printf("%s", s.c);
12.    }
```

- a) hello
 - b) Nothing
 - c) Varies
 - d) Compile time error
- [View Answer](#)

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    struct student
3.    {
4.        char *c;
5.        struct student *point;
6.    };
7.    void main()
8.    {
9.        struct student s;
10.        printf("%d", sizeof(s));
11.    }
```

- a) 5
- b) 9
- c) 8

d) 16

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.      struct student *point;
6.  };
7.  void main()
8.  {
9.      struct student s;
10.     struct student *m = &s;
11.     printf("%d", sizeof(student));
12. }
```

a) Compile time error

b) 8

c) 5

d) 16

[View Answer](#)

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
5.      char y;
6.      struct p *ptr;
7.  };
8.  int main()
```



```

9.    {
10.    struct p p = {1, 2, &p};
11.    printf("%d\n", p.ptr->x);
12.    return 0;
13.    }

```

- a) Compile time error
- b) Undefined behaviour
- c) 1
- d) 2

View Answer

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    typedef struct p *q;
3.    struct p
4.    {
5.        int x;
6.        char y;
7.        q ptr;
8.    };
9.    int main()
10.   {
11.       struct p p = {1, 2, &p};
12.       printf("%d\n", p.ptr->x);
13.       return 0;
14.   }

```

- a) Compile time error
- b) 1
- c) Undefined behaviour
- d) Address of p

View Answer

Answer: b

Explanation: None.

8. Presence of loop in a linked list can be tested by _____
- a) Traveling the list, if NULL is encountered no loop exists
 - b) Comparing the address of nodes by address of every other node
 - c) Comparing the the value stored in a node by a value in every other node
 - d) None of the mentioned

View Answer

Answer: b

Explanation: None.

Self-Referential Structures – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  typedef struct p *q;
3.  int main()
4.  {
5.      struct p
6.      {
7.          int x;
8.          char y;
9.          q ptr;
10.     };
11.     struct p p = {1, 2, &p};
12.     printf("%d\n", p.ptr->x);
13.     return 0;
14. }
```

- a) Compile time error
- b) 1
- c) Depends on the compiler
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
```

```

4.     typedef struct p *q;
5.     struct p
6.     {
7.         int x;
8.         char y;
9.         q ptr;
10.    };
11.    struct p p = {1, 2, &p};
12.    printf("%d\n", p.ptr->x);
13.    return 0;
14.    }

```

- a) Compile time error
 - b) 1
 - c) Depends on the compiler
 - d) Depends on the standard
- View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    typedef struct p *q;
3.    struct p
4.    {
5.        int x;
6.        char y;
7.        q ptr;
8.    };
9.    int main()
10.   {
11.       struct p p = {1, 2, &p};
12.       printf("%d\n", p.ptr->ptr->x);
13.       return 0;
14.   }

```

- a) Compile time error
- b) Segmentation fault
- c) Undefined behaviour
- d) 1

View Answer

Answer: d

Explanation: None.

4. The number of distinct nodes the following struct declaration can point to is _____

- 1. struct node
- 2. {
- 3. struct node *left;
- 4. struct node *centre;
- 5. struct node *right;
- 6. };

- a) 1
- b) 2
- c) 3
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

5. Which of the following is not possible regarding the structure variable?

- a) A structure variable pointing to itself
- b) A structure variable pointing to another structure variable of same type
- c) 2 different type of structure variable pointing at each other
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

6. Which of the following technique is faster for travelling in binary trees?

- a) Iteration
- b) Recursion
- c) Both Iteration and Recursion
- d) Depends from compiler to compiler

View Answer

Answer: b

Explanation: None.

7. Which of the following will stop the loop at the last node of a linked list in the following C code snippet?

```
1. struct node
2. {
3.     struct node *next;
4. };
```

a)

```
while (p != NULL)
{
    p = p->next;
}
```

b)

```
while (p->next != NULL)
{
    p = p->next;
}
```

c)

```
while (1)
{
    p = p->next;
    if (p == NULL)
        break;
}
```

d) All of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

Table Lookup – 1

1. What will be the output of the following C code?

```
1. #include <stdio.h>
2. struct student
3. {
4.     char a[5];
```

```
5.    };
6.    void main()
7.    {
8.        struct student s[] = {"hi", "hey"};
9.        printf("%c", s[0].a[1]);
10.   }
```

- a) h
- b) i
- c) e
- d) y

[View Answer](#)

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        char *a[3] = {"hello", "this"};
5.        printf("%s", a[1]);
6.    }
```

- a) hello
- b) Varies
- c) this
- d) Compile time error

[View Answer](#)

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    void main()
3.    {
4.        int lookup[100] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
5.        printf("%d", lookup[3]);
6.    }
```

- a) 2
- b) 4
- c) Compile time error
- d) 3

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *a[3][3] = {"hey", "hi", "hello"}, {"his", "her", "hell"}
5.      , {"hellos", "hi's", "hmm"}};
6.      printf("%s", a[1][1]);
7.  }
```

- a) her
- b) hi
- c) Compile time error
- d) hi's

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      char *name;
5.      struct p *next;
6.  };
7.  struct p *ptarray[10];
8.  int main()
9.  {
10.     struct p p;
11.     p->name = "xyz";
```

```

12.    p->next = NULL;
13.    ptrary[0] = &p;
14.    printf("%s\n", p->name);
15.    return 0;
16.    }

```

- a) Compile time error
- b) Segmentation fault/code crash
- c) xyz
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    struct p
3.    {
4.        char *name;
5.        struct p *next;
6.    };
7.    struct p *ptrary[10];
8.    int main()
9.    {
10.        struct p p;
11.        p.name = "xyz";
12.        p.next = NULL;
13.        ptrary[0] = &p;
14.        printf("%s\n", ptrary[0]->name);
15.        return 0;
16.    }

```

- a) Compile time error
- b) Segmentation fault
- c) Undefined behaviour
- d) xyz

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      char *name;
5.      struct p *next;
6.  };
7.  struct p *ptarray[10];
8.  int main()
9.  {
10.     struct p p, q;
11.     p.name = "xyz";
12.     p.next = NULL;
13.     ptarray[0] = &p;
14.     strcpy(q.name, p.name);
15.     ptarray[1] = &q;
16.     printf("%s\n", ptarray[1]->name);
17.     return 0;
18. }
```

- a) Compile time error
- b) Segmentation fault/code crash
- c) Depends on the compiler
- d) xyz

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      struct p
```

```

5.      {
6.          char *name;
7.          struct p *next;
8.      };
9.      struct p p, q;
10.     p.name = "xyz";
11.     p.next = NULL;
12.     ptrary[0] = &p;
13.     strcpy(q.name, p.name);
14.     ptrary[1] = &q;
15.     printf("%s\n", ptrary[1]->name);
16.     return 0;
17. }

```

- a) Compile time error
- b) Depends on the compiler
- c) Undefined behaviour
- d) xyz

View Answer

Answer: c

Explanation: None.

Table Lookup – 2

1. Which function is responsible for searching in the table? (For #define IN 1, the name IN and replacement text 1 are stored in a "table")

- a) findout(s);
- b) lookup(s);
- c) find(s);
- d) lookfor(s);

View Answer

Answer: b

Explanation: None.

2. Which algorithm is used for searching in the table?

- a) List search
- b) Informed search
- c) Hash search
- d) Adversarial search

View Answer

Answer: c

Explanation: None.

3. Which function is responsible for recording the name “s” and the replacement text “t” in a table?

- a) install(s, t);
- b) fix(s, t);
- c) setup(s, t);
- d) settle(s, t);

View Answer

Answer: a

Explanation: None.

4. Which of the following statement is true?

- a) Install function uses lookup
- b) lookup function uses install
- c) Install and lookup function work independently
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

5. What happens when install(s, t) finds that the name being installed is already present in the table?

- a) It doesn't modify the name in the table
- b) It modifies the name with new definition
- c) It modifies off the new definition has higher priority
- d) It creates a new table and add the new definition in it

View Answer

Answer: b

Explanation: None.

6. In what situation, install function returns NULL?

- a) When there is no memory for adding new name
- b) When the name to be defined is already present in the table
- c) Whenever a new name is added to the table
- d) All of the mentioned

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char a[];
5.  };
```

```

6.    void main()
7.    {
8.        struct student s;
9.        printf("%d", sizeof(struct student));
10.   }

```

a) Compile time error

b) 8

c) 1

d) Varies

[View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        struct p
5.        {
6.            char *name;
7.            struct p *next;
8.        };
9.        struct p *ptarray[10];
10.       struct p p, q;
11.       p.name = "xyz";
12.       p.next = NULL;
13.       ptarray[0] = &p;
14.       q.name = (char*)malloc(sizeof(char)*3);
15.       strcpy(q.name, p.name);
16.       q.next = &q;
17.       ptarray[1] = &q;
18.       printf("%s\n", ptarray[1]->next->next->name);
19.   }

```

- a) Compile time error
- b) Depends on the compiler
- c) Undefined behaviour
- d) xyz

View Answer

Answer: d

Explanation: None.

C Typedefs – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  typedef struct student
3.  {
4.      char *a;
5.  }stu;
6.  void main()
7.  {
8.      struct stu s;
9.      s.a = "hi";
10.     printf("%s", s.a);
11. }
```

- a) Compile time error
- b) Varies
- c) hi
- d) h

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  typedef struct student
3.  {
4.      char *a;
5.  }stu;
6.  void main()
```

```

7.    {
8.        struct student s;
9.        s.a = "hey";
10.    printf("%s", s.a);
11.    }

```

- a) Compile time error
- b) Varies
- c) he
- d) hey

View Answer

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    typedef int integer;
3.    int main()
4.    {
5.        int i = 10, *ptr;
6.        float f = 20;
7.        integer j = i;
8.        ptr = &j;
9.        printf("%d\n", *ptr);
10.    return 0;
11.    }

```

- a) Compile time error
- b) Undefined behaviour
- c) Depends on the standard
- d) 10

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int (*(x()))[2];

```

```

3.   typedef int (*(*ptr)())[2] ptrfoo;
4.   int main()
5.   {
6.       ptrfoo ptr1;
7.       ptr1 = x;
8.       ptr1();
9.       return 0;
10.  }
11.  int (*(x()))[2]
12.  {
13.      int (*ary)[2] = malloc(sizeof*ary);
14.      return &ary;
15.  }

```

- a) Compile time error
 - b) Nothing
 - c) Undefined behaviour
 - d) Depends on the standard
- View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int *(*(x()))[2];
3.   typedef int **(*ptrfoo())[2];
4.   int main()
5.   {
6.       ptrfoo ptr1;
7.       ptr1 = x;
8.       ptr1();
9.       return 0;
10.  }
11.  int *(*(x()))[2]
12.  {

```

```
13.    int (*ary)[2] = malloc(sizeof * ary);  
14.    return &ary;  
15.    }
```

- a) Compile time error
- b) Nothing
- c) Undefined behaviour
- d) Depends on the standard

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.    #include <stdio.h>  
2.    typedef struct p  
3.    {  
4.        int x, y;  
5.    };  
6.    int main()  
7.    {  
8.        p k1 = {1, 2};  
9.        printf("%d\\n", k1.x);  
10.   }
```

- a) Compile time error
- b) 1
- c) 0
- d) Depends on the standard

[View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.    #include <stdio.h>  
2.    typedef struct p  
3.    {  
4.        int x, y;  
5.    }k = {1, 2};
```



```

6.    int main()
7.    {
8.        p k1 = k;
9.        printf("%d\n", k1.x);
10.   }

```

- a) Compile time error
- b) 1
- c) 0
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    typedef struct p
3.    {C
4.        int x, y;
5.    }k;
6.    int main()
7.    {
8.        struct p p = {1, 2};
9.        k k1 = p;
10.    printf("%d\n", k1.x);
11.   }

```

- a) Compile time error
- b) 1
- c) 0
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

C Typedefs – 2

1. Which is the correct syntax to use typedef for struct?

a)

```
typedef struct temp
```

```
{  
    int a;  
}TEMP;
```

b)

```
typedef struct  
{  
    int a;  
}TEMP;
```

c)

```
struct temp  
{  
    int a;  
};  
  
typedef struct temp TEMP;
```

d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

2. Which option should be selected to work the following C expression?

```
string p = "HELLO";
```

- a) typedef char [] string;
- b) typedef char *string;
- c) typedef char [] string; and typedef char *string;
- d) Such expression cannot be generated in C

[View Answer](#)

Answer: b

Explanation: None.

3. Which of the given option is the correct method for initialization?

```
typedef char *string;
```

- a) *string *p = "Hello";
- b) string p = "Hello";
- c) *string p = 'A';
- d) Not more than one space should be given when using typedef

[View Answer](#)

Answer: b

Explanation: None.

4. Which of the following is false about typedef?

- a) typedef follow scope rules
- b) typedef defined substitutes can be redefined again. (Eg: typedef char a; typedef int a;)
- c) You cannot typedef a typedef with other term
- d) All of the mentioned

View Answer

Answer: b

Explanation: None.

5. Which of the following may create problem in the typedef program?

- a) ;
- b) printf/scanf
- c) Arithmetic operators
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

6. typedef int (*PFI)(char *, char *)creates _____

- a) type PFI, for pointer to function (of two char * arguments) returning int
- b) error
- c) type PFI, function (of two char * arguments) returning int
- d) type PFI, for pointer

View Answer

Answer: a

Explanation: None.

7. What is typedef declaration?

- a) Does not create a new type
- b) It merely adds a new name for some existing type
- c) Does not create a new type, It merely adds a new name for some existing type
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

8. What will be the output of the following C code?

1. #include <stdio.h>
2. typedef struct student
3. {
4. char *a;
5. }stu;

```

6.    void main()
7.    {
8.        stu s;
9.        s.a = "hi";
10.     printf("%s", s.a);
11.    }

```

- a) Compile time error
- b) Varies
- c) hi
- d) h

View Answer

Answer: a

Explanation: None.

C Unions – 1

1. The size of a union is determined by the size of the _____

- a) First member in the union
- b) Last member in the union
- c) Biggest member in the union
- d) Sum of the sizes of all members

View Answer

Answer: c

Explanation: None.

2. Which member of the union will be active after REF LINE in the following C code?

```

1.    #include <stdio.h>
2.    union temp
3.    {
4.        int a;
5.        float b;
6.        char c;
7.    };
8.    union temp s = {1,2.5,'A'}; //REF LINE

```

- a) a
- b) b
- c) c
- d) Such declaration are illegal

View Answer

Answer: a

Explanation: None.

3. What would be the size of the following union declaration? (Assuming size of double = 8, size of int = 4, size of char = 1)

```
1.  #include <stdio.h>
2.  union uTemp
3.  {
4.      double a;
5.      int b[10];
6.      char c;
7.  }u;
```

- a) 4
- b) 8
- c) 40
- d) 80

View Answer

Answer: c

Explanation: None.

4. What type of data is held by variable u in the following C code?

```
1.  #include <stdio.h>
2.  union u_tag
3.  {
4.      int ival;
5.      float fval;
6.      char *sval;
7.  } u;
```

- a) Will be large enough to hold the largest of the three types;
- b) Will be large enough to hold the smallest of the three types;
- c) Will be large enough to hold the all of the three types;
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

5. Members of a union are accessed as _____

- a) union-name.member
- b) union-pointer->member

c) both union-name.member & union-pointer->member

d) none of the mentioned

View Answer

Answer: c

Explanation: None.

6. In the following C code, we can access the 1st character of the string sval by using _____

```
1.  #include <stdio.h>
2.  struct
3.  {
4.      char *name;
5.      union
6.      {
7.          char *sval;
8.      } u;
9.  } symtab[10];
```

a) *symtab[i].u.sval

b) symtab[i].u.sval[0].

c) You cannot have union inside structure

d) Both *symtab[i].u.sval & symtab[i].u.sval[0].

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code (Assuming size of int and float is 4)?

```
1.  #include <stdio.h>
2.  union
3.  {
4.      int ival;
5.      float fval;
6.  } u;
7.  void main()
8.  {
9.      printf("%d", sizeof(u));
10. }
```

- a) 16
- b) 8
- c) 4
- d) 32

View Answer

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  union stu
4.  {
5.      int ival;
6.      float fval;
7.  };
8.  void main()
9.  {
10.     union stu r;
11.     r.ival = 5;
12.     printf("%d", r.ival);
13. }
```

- a) 9
- b) Compile time error
- c) 16
- d) 5

View Answer

Answer: d

Explanation: None.

C Unions – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  union
4.  {
5.      int x;
6.      char y;
```

```

6.    }p;
7.    int main()
8.    {
9.        p.x = 10;
10.    printf("%d\n", sizeof(p));
11.    }

```

- a) Compile time error
- b) sizeof(int) + sizeof(char)
- c) Depends on the compiler
- d) sizeof(int)

View Answer

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    union
3.    {
4.        int x;
5.        char y;
6.    }p;
7.    int main()
8.    {
9.        p.y = 60;
10.    printf("%d\n", sizeof(p));
11.    }

```

- a) Compile time error
- b) sizeof(int) + sizeof(char)
- c) Depends on the compiler
- d) sizeof(char)

View Answer

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```

1.    #include <stdio.h>

```



```

2.    union p
3.    {
4.        int x;
5.        char y;
6.    };
7.    int main()
8.    {
9.        union p p, b;
10.       p.y = 60;
11.       b.x = 12;
12.       printf("%d\n", p.y);
13.    }

```

- a) Compile time error
- b) Depends on the compiler
- c) 60
- d) Undefined behaviour

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    union p
3.    {
4.        int x;
5.        char y;
6.    }k = {1, 97};
7.    int main()
8.    {
9.        printf("%d\n", k.y);
10.    }

```

- a) Compile time error
- b) 97
- c) a

d) 1

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  union p
3.  {
4.      int x;
5.      char y;
6.  }k = {.y = 97};
7.  int main()
8.  {
9.      printf("%d\n", k.y);
10. }
```

a) Compile time error

b) 97

c) a

d) Depends on the standard

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  union p
3.  {
4.      int x;
5.      float y;
6.  };
7.  int main()
8.  {
9.      union p p, b;
10.     p.x = 10;
```

11. printf("%f\n", p.y);

12. }

- a) Compile time error
- b) Implementation dependent
- c) 10.000000
- d) 0.000000

View Answer

Answer: b

Explanation: None.

7. Which of the following share a similarity in syntax?

1. Union, 2. Structure, 3. Arrays and 4. Pointers

- a) 3 and 4
- b) 1 and 2
- c) 1 and 3
- d) 1, 3 and 4

View Answer

Answer: b

Explanation: None.

8. What will be the output of the following C code? (Assuming size of char = 1, int = 4, double = 8)

```
1.   #include <stdio.h>
2.   union utemp
3.   {
4.       int a;
5.       double b;
6.       char c;
7.   }u;
8.   int main()
9.   {
10.      u.c = 'A';
11.      u.a = 1;
12.      printf("%d", sizeof(u));
13.  }
```

- a) 1
- b) 4
- c) 8

d) 13

[View Answer](#)

Answer: c

Explanation: None.

C Bit-fields – 1

1. What is the correct syntax to initialize bit-fields in an structure?

a)

```
struct temp
{
    unsigned int a : 1;
};
```

b)

```
struct temp
{
    unsigned int a = 1;
};
```

c)

```
struct temp
{
    unsigned float a : 1;
};
```

d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

2. Which of the following data types are accepted while declaring bit-fields?

a) char

b) float

c) double

d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

3. Which of the following reduces the size of a structure?

a) union

- b) bit-fields
- c) malloc
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

4. For what minimum value of x in a 32-bit Linux OS would make the size of s equal to 8 bytes?

1. struct temp
2. {
3. int a : 13;
4. int b : 8;
5. int c : x;
6. }s;

- a) 4
- b) 8
- c) 12
- d) 32

View Answer

Answer: c

Explanation: None.

5. Calculate the % of memory saved when bit-fields are used for the following C structure as compared to with-out use of bit-fields for the same structure? (Assuming size of int = 4)

1. struct temp
2. {
3. int a : 1;
4. int b : 2;
5. int c : 4;
6. int d : 4;
7. }s;

- a) 25%
- b) 33.3%
- c) 50%
- d) 75%

View Answer

Answer: d

Explanation: None.

6. In the following declaration of bit-fields, the constant-expression specifies _____

struct-declarator:

declarator

type-specifier declarator opt : constant-expression

- a) The width of the field in bits
- b) Nothing
- c) The width of the field in bytes
- d) Error

[View Answer](#)

Answer: a

Explanation: None.

7. In the following declaration of bit-fields, the constant-expression must be _____

struct-declarator:

declarator

type-specifier declarator opt : constant-expression

- a) Any type
- b) Nothing
- c) Integer value
- d) Nonnegative integer value

[View Answer](#)

Answer: d

Explanation: None.

8. Which of the following is not allowed?

- a) Arrays of bit fields
- b) Pointers to bit fields
- c) Functions returning bit fields
- d) None of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

9. Bit fields can only be declared as part of a structure.

- a) false
- b) true
- c) Nothing
- d) Varies

[View Answer](#)

Answer: b

Explanation: None.

10. What is the order for the following C declarations?

short a : 17;

int long y : 33;

- a) Legal, legal
- b) Legal, illegal
- c) Illegal, illegal
- d) Illegal, legal

View Answer

Answer: c

Explanation: None.

C Bit-fields – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    struct p
3.    {
4.        char x : 2;
5.        int y : 2;
6.    };
7.    int main()
8.    {
9.        struct p p;
10.       p.x = 2;
11.       p.y = 1;
12.       p.x = p.x & p.y;
13.       printf("%d\n", p.x);
14.    }
```

- a) 0
- b) Compile time error
- c) Undefined behaviour
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```

2.    union u
3.    {
4.        struct p
5.        {
6.            unsigned char x : 2;
7.            unsigned int y : 2;
8.        };
9.        int x;
10.   };
11. int main()
12. {
13.     union u u;
14.     u.p.x = 2;
15.     printf("%d\n", u.p.x);
16. }

```

- a) Compile time error
- b) Undefined behaviour
- c) Depends on the standard
- d) 2

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    union u
3.    {
4.        struct
5.        {
6.            unsigned char x : 2;
7.            unsigned int y : 2;
8.        }p;
9.        int x;
10.   };

```



```

11. int main()
12. {
13.     union u u;
14.     u.p.x = 2;
15.     printf("%d\n", u.p.x);
16. }

```

- a) Compile time error
- b) 2
- c) Undefined behaviour
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  union u
3.  {
4.      struct
5.      {
6.          unsigned char x : 2;
7.          unsigned int y : 2;
8.      }p;
9.      int x;
10. };
11. int main()
12. {
13.     union u u.p.x = 2;
14.     printf("%d\n", u.p.x);
15. }

```

- a) Compile time error
- b) 2
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  union u
3.  {
4.      struct
5.      {
6.          unsigned char x : 2;
7.          unsigned int y : 2;
8.      }p;
9.      int x;
10. };
11. int main()
12. {
13.     union u u = {2};
14.     printf("%d\n", u.p.x);
15. }
```

a) Compile time error

b) 2

c) Depends on the standard

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  union u
3.  {
4.      struct
5.      {
6.          unsigned char x : 2;
7.          unsigned int y : 2;
```

```

8.     }p;
9.     int x;
10.    };
11.    int main()
12.    {
13.        union u u.p = {2};
14.        printf("%d\n", u.p.x);
15.    }

```

- a) Compile time error
- b) 2
- c) Undefined behaviour
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.
3.    struct p
4.    {
5.        unsigned int x : 2;
6.        unsigned int y : 2;
7.    };
8.    int main()
9.    {
10.        struct p p;
11.        p.x = 3;
12.        p.y = 1;
13.        printf("%d\n", sizeof(p));
14.    }

```

- a) Compile time error
- b) Depends on the compiler
- c) 2
- d) 4

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      unsigned int x : 2;
5.      unsigned int y : 2;
6.  };
7.  int main()
8.  {
9.      struct p p;
10.     p.x = 3;
11.     p.y = 4;
12.     printf("%d\n", p.y);
13. }
```

a) 0

b) 4

c) Depends on the compiler

d) 2

[View Answer](#)

Answer: a

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      unsigned int x : 7;
5.      unsigned int y : 2;
6.  };
7.  int main()
8.  {
9.      struct p p;
```

```
10.    p.x = 110;
11.    p.y = 2;
12.    printf("%d\n", p.x);
13. }
```

- a) Compile time error
- b) 110
- c) Depends on the standard
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

10. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    struct p
3.    {
4.        unsigned int x : 1;
5.        unsigned int y : 1;
6.    };
7.    int main()
8.    {
9.        struct p p;
10.       p.x = 1;
11.       p.y = 2;
12.       printf("%d\n", p.y);
13. }
```

- a) 1
- b) 2
- c) 0
- d) Depends on the compiler

View Answer

Answer: c

Explanation: None.

Multiple Choice Questions on C Input and Output

Standard Input & Output – 1

1. Which among the following is the odd one out?

- a) printf
 - b) fprintf
 - c) putchar
 - d) scanf
- View Answer

Answer: d

Explanation: None.

2. For a typical program, the input is taken using _____

- a) scanf
 - b) Files
 - c) Command-line
 - d) All of the mentioned
- View Answer

Answer: d

Explanation: None.

3. What does the following command line signify?

prog1|prog2

- a) It runs prog1 first, prog2 second
 - b) It runs prog2 first, prog1 second
 - c) It runs both the programs, pipes output of prog1 to input of prog2
 - d) It runs both the programs, pipes output of prog2 to input of prog1
- View Answer

Answer: c

Explanation: None.

4. What is the default return-type of getchar()?

- a) char
 - b) int
 - c) char *
 - d) reading character doesn't require a return-type
- View Answer

Answer: b

Explanation: None.

5. What is the value of EOF?

- a) -1
- b) 0
- c) 1

d) 10

View Answer

Answer: a

Explanation: None.

6. What is the use of `getchar()`?

a) The next input character each time it is called

b) EOF when it encounters end of file

c) The next input character each time it is called EOF when it encounters end of file

d) None of the mentioned

View Answer

Answer: c

Explanation: None.

7. Which of the following statement is true?

a) The symbolic constant EOF is defined in `<stdio.h>`

b) The value is -1

c) The symbolic constant EOF is defined in `<stdio.h>` & value is -1

d) Only value is -1

View Answer

Answer: c

Explanation: None.

8. What is the return value of `putchar()`?

a) The character written

b) EOF if an error occurs

c) Nothing

d) Both character written & EOF if an error occurs

View Answer

Answer: d

Explanation: None.

Standard Input & Output – 2

1. Which is not true about function `tolower`?

a) The function `tolower` is defined in `<ctype.h>`

b) Converts an uppercase letter to lowercase

c) Returns other characters untouched

d) None of the mentioned

View Answer

Answer: d

Explanation: None.

2. What will be the output of the following C code?

1. `#include <stdio.h>`

2. `int main()`

```
3.  {  
4.      char c = '❖';  
5.      putchar(c);  
6.  }
```

- a) Compile time error
- b) Nothing
- c) 0
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

3. putchar(c) function/macro always outputs character c to the _____

- a) screen
- b) standard output
- c) depends on the compiler
- d) depends on the standard

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code if following commands are used to run (considering myfile exists)?

```
1.  gcc -o test test.c  
2.  ./test < myfile  
3.  
4.  #include <stdio.h>  
5.  int main()  
6.  {  
7.      char c = 'd';  
8.      putchar(c);  
9.  }
```

- a) Compile time error (after first command)
- b) d in the myfile file
- c) d on the screen
- d) Undefined behaviour

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code if following commands are used to run (considering myfile exists)?

```
1. gcc -otest test.c
2. ./test > myfile
3.
4. #include <stdio.h>
5. int main(int argc, char **argv)
6. {
7.     char c = 'd';
8.     putchar(c);
9.     printf(" %d\n", argc);
10. }
```

- a) d 2 in myfile
- b) d 1 in myfile
- c) d in myfile and 1 in screen
- d) d in myfile and 2 in screen

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code if following commands are used to run and if myfile does not exist?

```
1. gcc -o test test.c
2. ./test > myfile
3.
4. #include <stdio.h>
5. int main(int argc, char **argv)
6. {
7.     char c = 'd';
8.     putchar(c);
9.     printf(" %d\n", argc);
10. }
```

- a) d 2 in myfile
- b) d 1 in myfile
- c) Depends on the system

d) Depends on the standard

[View Answer](#)

Answer: b

Explanation: None.

7. The statement `prog < infile` causes _____

a) prog to read characters from infile

b) prog to write characters to infile

c) infile to read characters from prog instead

d) nothing

[View Answer](#)

Answer: a

Explanation: None.

Formatted Output – 1

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 10, j = 2;
5.        printf("%d\n", printf("%d %d ", i, j));
6.    }
```

a) Compile time error

b) 10 2 4

c) 10 2 2

d) 10 2 5

[View Answer](#)

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 10, j = 3;
5.        printf("%d %d %d", i, j);
6.    }
```

a) Compile time error

b) 10 3

c) 10 3 some garbage value

d) Undefined behaviour

[View Answer](#)

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 10, j = 3, k = 3;
5.      printf("%d %d ", i, j, k);
6.  }
```

a) Compile time error

b) 10 3 3

c) 10 3

d) 10 3 somegarbage value

[View Answer](#)

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *s = "myworld";
5.      int i = 9;
6.      printf("%*s", i, s);
7.  }
```

a) myworld

b) myworld(note: spaces to the left of myworld)

c) myworld (note:followed by two spaces after myworld)

d) Undefined

[View Answer](#)

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```

2.   int main(int argc, char** argv)
3.   {
4.       char *s = "myworld";
5.       int i = 3;
6.       printf("%10.*s", i, s);
7.   }

```

- a) myw(note:7 spaces before myw)
- b) myworld(note:2 spaces before myworld)
- c) myworld (note:2 spaces after myworld)
- d) myw(note:6 spaces after myw)

View Answer

Answer: a

Explanation: In the format represented by “%10.*s”, the width of the string will be 10 spaces which is aligned to the right, by default. Since we have asterisk (*) after the precision dot (.), the value of precision will be the value stored in the variable i. The value of i is 3, so this signifies max length of the string as 3 characters. So, the final formatted output will be a 10 character output with 3 characters “myw” printed with right alignment and the 1st 7 characters will be simply space characters.

6. What is the difference between %e and %g?

- a) %e output formatting depends on the argument and %g always formats in the format [-]m.dddddd or [-]m.dddddE[+|-]xx where no.of ds are optional
- b) %e always formats in the format [-]m.dddddd or [-]m.dddddE[+|-]xx where no.of ds are optional and output formatting depends on the argument
- c) No differences
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

7. Escape sequences are prefixed with _____

- a) %
- b) /
- c) ”
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

8. What is the purpose of sprintf?

- a) It prints the data into stdout
- b) It writes the formatted data into a string
- c) It writes the formatted data into a file

d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

9. The syntax to print a % using printf statement can be done by _____

a) %

b) \%

c) '%'

d) %%

[View Answer](#)

Answer: d

Explanation: None.

Formatted Output – 2

1. What is the meaning of the following C statement?

```
printf("%10s", state);
```

a) 10 spaces before the string state is printed

b) Print empty spaces if the string state is less than 10 characters

c) Print the last 10 characters of the string

d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

2. What are the Properties of the first argument of a printf() functions?

a) It is defined by a user

b) It keeps the record of the types of arguments that will follow

c) There may no be first argument

d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 10, j = 2;
5.        printf("%d\n", printf("%d %d ", i, j));
6.    }
```

- a) Compile time error
- b) 10 2 4
- c) 10 2 2
- d) 10 2 5

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 10, j = 3;
5.        printf("%d %d %d", i, j);
6.    }
```

- a) Compile time error
- b) 10 3
- c) 10 3 some garbage value
- d) Undefined behaviour

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        int i = 10, j = 3, k = 3;
5.        printf("%d %d ", i, j, k);
6.    }
```

- a) Compile time error
- b) 10 3 3
- c) 10 3
- d) 10 3 somegarbage value

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *s = "myworld";
5.      int i = 9;
6.      printf("%*s", i, s);
7.  }

```

- a) myworld
- b) myworld(note: spaces to the left of myworld)
- c) myworld (note:followed by two spaces after myworld)
- d) Undefined

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main(int argc, char **argv)
3.  {
4.      char *s = "myworld";
5.      int i = 3;
6.      printf("%10.*s", i, s);
7.  }

```

- a) myw(note:7 spaces before myw)
- b) myworld(note:2 spaces before myworld)
- c) myworld (note:2 spaces after myworld)
- d) myw(note:6 spaces after myworld)

View Answer

Answer: a

Explanation: In the format represented by “%10.*s”, the width of the string will be 10 spaces which is aligned to the right, by default. Since we have asterisk (*) after the precision dot (.), the value of precision will be the value stored in the variable i. The value of i is 3, so this signifies max length of the string as 3 characters. So, the final formatted output will be a 10 character output with 3 characters “myw” printed with right alignment and the 1st 7 characters will be simply space characters.

8. What is the difference between %e and %g?

- a) %e output formatting depends on the argument and %g always formats in the format [-]m.dddddd or [-]m.dddddE[+|-]xx where no.of ds are optional

- b) %e always formats in the format [-]m.dddddd or [-]m.dddddE[+|-]xx where no.of ds are optional and output formatting depends on the argument
- c) No differences
- d) Depends on the standard

[View Answer](#)

Answer: b

Explanation: None.

Variable Length Argument – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <stdarg.h>
3.  void func(int, ...);
4.  int main()
5.  {
6.      func(2, 3, 5, 7, 11, 13);
7.      return 0;
8.  }
9.  void func(int n, ...)
10. {
11.     int number, i = 0;
12.     va_list start;
13.     va_start(start, n);
14.     while (i != 3)
15.     {
16.         number = va_arg(start, int);
17.         i++;
18.     }
19.     printf("%d", number);
20. }
```

- a) 3
- b) 5
- c) 7
- d) 11

[View Answer](#)

Answer: c

Explanation: None.

2. Which of the following function with ellipsis are illegal?

- a) void func(...);
- b) void func(int, ...);
- c) void func(int, int, ...);
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

3. Which of the following data-types are promoted when used as a parameter for an ellipsis?

- a) char
- b) short
- c) int
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

4. Which header file includes a function for variable number of arguments?

- a) stdlib.h
- b) stdarg.h
- c) ctype.h
- d) both stdlib.h and stdarg.h

View Answer

Answer: b

Explanation: None.

5. Which of the following macro extracts an argument from the variable argument list (ie ellipsis) and advance the pointer to the next argument?

- a) va_list
- b) va_arg
- c) va_end
- d) va_start

View Answer

Answer: b

Explanation: None.

6. The type va_list in an argument list is used _____

- a) To declare a variable that will refer to each argument in turn;
- b) For cleanup
- c) To create a list
- d) There is no such type

View Answer

Answer: a

Explanation: None.

7. In a variable length argument function, the declaration “...” can _____

- a) Appear anywhere in the function declaration
- b) Only appear at the end of an argument list
- c) Nothing
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

8. Each call of va_arg _____

- a) Returns one argument
- b) Steps va_list variable to the next
- c) Returns one argument & Steps va_list variable to the next
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

Variable Length Argument – 2

1. The standard header _____ is used for variable list arguments (...) in C.

- a) <stdio.h>
- b) <stdlib.h>
- c) <math.h>
- d) <stdarg.h>

View Answer

Answer: d

Explanation: None.

2. What is the purpose of va_end?

- a) Cleanup is necessary
- b) Must be called before the program returns
- c) Cleanup is necessary & Must be called before the program returns
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

3. What will be the output of the following C code?

- 1. #include <stdio.h>
- 2. int f(char chr, ...);
- 3. int main()
- 4. {

```

5.     char c = 97;
6.     f(c);
7.     return 0;
8.     }
9.     int f(char c, ...)
10.    {
11.        printf("%c\n", c);
12.    }

```

- a) Compile time error
- b) Undefined behaviour
- c) 97
- d) a

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     #include <stdarg.h>
3.     int f(...);
4.     int main()
5.     {
6.         char c = 97;
7.         f(c);
8.         return 0;
9.     }
10.    int f(...)
11.    {
12.        va_list li;
13.        char c = va_arg(li, char);
14.        printf("%c\n", c);
15.    }

```

- a) Compile time error
- b) Undefined behaviour

c) 97

d) a

[View Answer](#)

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <stdarg.h>
3.  int f(char c, ...);
4.  int main()
5.  {
6.      char c = 97, d = 98;
7.      f(c, d);
8.      return 0;
9.  }
10. int f(char c, ...)
11. {
12.     va_list li;
13.     va_start(li, c);
14.     char d = va_arg(li, char);
15.     printf("%c\n", d);
16.     va_end(li);
17. }
```

a) Compile time error

b) Undefined behaviour

c) a

d) b

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <stdarg.h>
3.  int f(char c, ...);
```

```

4.   int main()
5.   {
6.       char c = 97, d = 98;
7.       f(c, d);
8.       return 0;
9.   }
10.  int f(char c, ...)
11.  {
12.      va_list li;
13.      va_start(li, c);
14.      char d = va_arg(li, int);
15.      printf("%c\n", d);
16.      va_end(li);
17.  }

```

- a) Compile time error
- b) Undefined behaviour
- c) a
- d) b

View Answer

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   #include <stdarg.h>
3.   int f(int c, ...);
4.   int main()
5.   {
6.       int c = 97;
7.       float d = 98;
8.       f(c, d);
9.       return 0;
10.  }
11.  int f(int c, ...)

```

```
12.  {
13.    va_list li;
14.    va_start(li, c);
15.    float d = va_arg(li, float);
16.    printf("%f\n", d);
17.    va_end(li);
18. }
```

- a) Compile time error
- b) Undefined behaviour
- c) 97.000000
- d) 98.000000

[View Answer](#)

Answer: b

Explanation: None.

Formatted Input – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  int main()
4.  {
5.      int n;
6.      scanf("%d", n);
7.      printf("%d\n", n);
8.      return 0;
9.  }
```

- a) Compilation error
- b) Undefined behavior
- c) Whatever user types
- d) Depends on the standard

[View Answer](#)

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  int main()
```

```

3.  {
4.      char *n;
5.      scanf("%s", n);
6.      return 0;
7.  }

```

- a) Compilation error
- b) Undefined behavior
- c) Nothing
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      char n[] = "hello\nworld!";
5.      char s[13];
6.      sscanf(n, "%s", s);
7.      printf("%s\n", s);
8.      return 0;
9.  }

```

- a) hellonworld!
- b)

hello

world!

- c) hello
- d) hello world!

View Answer

Answer: c

Explanation: The array **n** contains a string which has a newline character in between the strings "hello" and "world". A newline character is considered as a whitespace character for inputs for the `scanf()`, `sscanf()` and `fscanf()` functions. So, the `sscanf()` function will only copy upto the string "hello" into the array **s**. Hence, the output of the `printf()` function be only the string "hello".

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      short int i;
5.      scanf("%hd", &i);
6.      printf("%hd", i);
7.      return 0;
8.  }
```

- a) Compilation error
- b) Undefined behavior
- c) Whatever user types
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      short int i;
5.      scanf("%*d", &i);
6.      printf("%hd", i);
7.      return 0;
8.  }
```

- a) Compilation error
- b) Somegarbage value
- c) Whatever user types
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
```



```
3.  {
4.      short int i;
5.      scanf("%*hd", &i);
6.      printf("%hd", i);
7.      return 0;
8.  }
```

- a) Compilation error
- b) Somegarbage value
- c) Whatever user types
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      short int i;
5.      scanf("%h*d", &i);
6.      printf("%hd", i);
7.      return 0;
8.  }
```

- a) Compilation error
- b) Undefined behavior
- c) Somegarbage value
- d) Depends on the standard.

View Answer

Answer: a

Explanation: None.

8. Which of the following is NOT a delimiter for an input in scanf?

- a) Enter
- b) Space
- c) Tab
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

9. If the conversion characters of int d, i, o, u and x are preceded by h, it indicates?

- a) A pointer to int
- b) A pointer to short
- c) A pointer to long
- d) A pointer to char

View Answer

Answer: b

Explanation: None.

Formatted Input – 2

1. Which of the following doesn't require an & for the input in scanf()?

- a) char name[10];
- b) int name[10];
- c) float name[10];
- d) all of the mentioned

View Answer

Answer: a

Explanation: None.

2. Which of the following is an invalid method for input?

- a) scanf("%d%d%d",&a, &b, &c);
- b) scanf("%d %d %d", &a, &b, &c);
- c) scanf("Three values are %d %d %d",&a,&b,&c);
- d) none of the mentioned

View Answer

Answer: d

Explanation: None.

3. Which of the following represents the function for scanf()?

- a) void scanf(char *format, ...)
- b) int scanf(char *format, ...)
- c) char scanf(int format, ...)
- d) char *scanf(char *format, ...)

View Answer

Answer: b

Explanation: None.

4. What does scanf() function return?

- a) Number of successfully matched and assigned input items
- b) Nothing
- c) Number of characters properly printed
- d) Error

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int n;
5.      scanf("%d", n);
6.      printf("%d", n);
7.  }
```

- a) Prints the number that was entered
- b) Segmentation fault
- c) Nothing
- d) Varies

[View Answer](#)

Answer: b

Explanation: scanf() expects a pointer/address to be passed. In the given code, we are simply passing a value which is uninitialized. This results in segmentation fault.

6. What will be the output of the following C statement?

```
int sscanf(char *string, char *format, arg1, arg2, ...)
```

- a) Scans the string according to the format in format and stores the resulting values through arg1, arg2, etc
- b) The arguments arg1, arg2 etc must be pointers
- c) Scans the string according to the format in format and stores the resulting values through arg1, arg2, etc, those arguments arg1, arg2 etc must be pointers
- d) None of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

7. The conversion characters d, i, o, u, and x may be preceded by h in scanf() to indicate?

- a) A pointer to short
- b) A pointer to long
- c) Nothing
- d) Error

[View Answer](#)

Answer: a

Explanation: None.

8. What will be the output of the following C code (when 4 and 5 are entered)?

```
1.  #include <stdio.h>
2.  void main()
```

```

3.  {
4.      int m, n;
5.      printf("enter a number");
6.      scanf("%d", &n);
7.      scanf("%d", &m);
8.      printf("%d\t%d\n", n, m);
9.  }

```

- a) Error
- b) 4 junkvalue
- c) Junkvalue 5
- d) 4 5

View Answer

Answer: d

Explanation: None.

File Access – 1

1. What are the first and second arguments of fopen?

- a) A character string containing the name of the file & the second argument is the mode
- b) A character string containing the name of the user & the second argument is the mode
- c) A character string containing file pointer & the second argument is the mode
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

2. For binary files, a ____ must be appended to the mode string.

- a) Nothing
- b) "b"
- c) "binary"
- d) "01"

View Answer

Answer: b

Explanation: None.

3. What will fopen will return, if there is any error while opening a file?

- a) Nothing
- b) EOF
- c) NULL
- d) Depends on compiler

View Answer

Answer: c

Explanation: None.

4. What is the return value of `getc()`?

- a) The next character from the stream is not referred by file pointer
- b) EOF for end of file or error
- c) Nothing
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

5. When a C program is started, O.S environment is responsible for opening file and providing pointer for that file?

- a) Standard input
- b) Standard output
- c) Standard error
- d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

6. In C language, FILE is of which data type?

- a) int
- b) char *
- c) struct
- d) None of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

7. What is meant by 'a' in the following C operation?

```
fp = fopen("Random.txt", "a");
```

- a) Attach
- b) Append
- c) Apprehend
- d) Add

[View Answer](#)

Answer: b

Explanation: None.

8. Which of the following mode argument is used to truncate?

- a) a
- b) f
- c) w
- d) t

[View Answer](#)

Answer: c

Explanation: None.

9. Which type of files can't be opened using fopen()?

- a) .txt
- b) .bin
- c) .c
- d) none of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

File Access – 2

1. Which of the following fopen() statements are illegal?

- a) fp = fopen("abc.txt", "r");
- b) fp = fopen("/home/user1/abc.txt", "w");
- c) fp = fopen("abc", "w");
- d) none of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

2. What does the following segment of C code do?

```
fprintf(fp, "Copying!");
```

- a) It writes "Copying!" into the file pointed by fp
- b) It reads "Copying!" from the file and prints on display
- c) It writes as well as reads "Copying!" to and from the file and prints it
- d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

3. What is FILE reserved word?

- a) A structure tag declared in stdio.h
- b) One of the basic data types in c
- c) Pointer to the structure defined in stdio.h
- d) It is a type name defined in stdio.h

[View Answer](#)

Answer: d

Explanation: None.

4. What will be the output of the following C code?

1. `#include <stdio.h>`
2. `int main()`
3. `{`
4. `FILE *fp = stdin;`

```
5.     int n;  
6.     fprintf(fp, "%d", 45);  
7. }
```

- a) Compilation error
- b) 45
- c) Nothing
- d) Depends on the standard

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.     #include <stdio.h>  
2.     #include <stdlib.h>  
3.     int main()  
4.     {  
5.         FILE *fp = stdout;  
6.         int n;  
7.         fprintf(fp, "%d", 45);  
8.     }
```

- a) Compilation error
- b) 45
- c) Nothing
- d) Depends on the standard

[View Answer](#)

Answer: b

Explanation: None.

6. stdout, stdin and stderr are _____

- a) File pointers
- b) File descriptors
- c) Streams
- d) Structure

[View Answer](#)

Answer: a

Explanation: None.

7. Which of the following statements about stdout and stderr are true?

- a) Same
- b) Both connected to screen always
- c) Both connected to screen by default

d) stdout is line buffered but stderr is unbuffered

[View Answer](#)

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      FILE *fp = stdout;
5.      int n;
6.      fprintf(fp, "%d ", 45);
7.      fprintf(stderr, "%d ", 65);
8.      return 0;
9.  }
```

a) 45 65

b) 65 45

c) 65

d) Compilation error

[View Answer](#)

Answer: b

Explanation: None.

9. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      FILE *fp = stdout;
5.      int n;
6.      fprintf(fp, "%d\n", 45);
7.      fprintf(stderr, "%d ", 65);
8.      return 0;
9.  }
```

a) 45 65

b) 65 45

c) 65

d) Compilation error

[View Answer](#)

Answer: a

Explanation: None.

10. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      FILE *fp = stdout;
5.      int n;
6.      fprintf(fp, "%d ", 45);
7.      fflush(stdout);
8.      fprintf(stderr, "%d", 65);
9.      return 0;
10. }
```

a) 45 65

b) 65 45

c) 45

d) Compilation error

[View Answer](#)

Answer: a

Explanation: None.

Error Handling – 1

1. What is the output of the following C code if there is no error in stream fp?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      FILE *fp;
5.      fp = fopen("newfile", "w");
6.      printf("%d\n", ferror(fp));
7.      return 0;
8.  }
```

a) Compilation error

b) 0

- c) 1
 - d) Any nonzero value
- View Answer

Answer: b
Explanation: None.

2. Within main, return expr statement is equivalent to _____
- a) abort(expr)
 - b) exit(expr)
 - c) ferror(expr)
 - d) none of the mentioned
- View Answer

Answer: b
Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      FILE *fp;
5.      char c;
6.      int n = 0;
7.      fp = fopen("newfile1.txt", "r");
8.      while (!feof(fp))
9.      {
10.         c = getc(fp);
11.         putc(c, stdout);
12.     }
13. }
```

- a) Compilation error
 - b) Prints to the screen content of newfile1.txt completely
 - c) Prints to the screen some contents of newfile1.txt
 - d) None of the mentioned
- View Answer

Answer: d
Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
```

```

2.   int main()
3.   {
4.       FILE *fp = stdout;
5.       stderr = fp;
6.       fprintf(stderr, "%s", "hello");
7.   }

```

- a) Compilation error
- b) hello
- c) Undefined behaviour
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   int main()
3.   {
4.       char buf[12];
5.       stderr = stdin;
6.       fscanf(stderr, "%s", buf);
7.       printf("%s\n", buf);
8.   }

```

- a) Compilation error
- b) Undefined behaviour
- c) Whatever user types
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

6. stderr is similar to?

- a) stdin
- b) stdout
- c) both stdout and stdin
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

7. What happens when we use the following C statement?

```
fprintf(stderr, "error: could not open file");
```

- a) The diagnostic output is directly displayed in the output
- b) The diagnostic output is pipelined to the output file
- c) The line which caused error is compiled again
- d) The program is immediately aborted

View Answer

Answer: a

Explanation: None.

8. Which of the following function can be used to terminate the main function from another function safely?

- a) return(expr);
- b) exit(expr);
- c) abort();
- d) both exit(expr); and abort();

View Answer

Answer: b

Explanation: None.

Error Handling – 2

1. Which of the following causes an error?

- a) Trying to read a file that doesn't exist
- b) Inability to write data in a file
- c) Failure to allocate memory with the help of malloc
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

2. What is the purpose of the C function?

```
int ferror(FILE *fp)
```

- a) They check for input errors
- b) They check for output errors
- c) They check for all types of errors
- d) They check for error in accessing the file

View Answer

Answer: b

Explanation: None.

3. stderr is similar to?

- a) stdin
- b) stdout
- c) Both stdout and stdin

d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: stderr is not exactly the same as stdout, but similar in the sense that both puts the output or error to the monitor.

4. What will be the output of the following C statement?

```
fprintf(stderr, "error: could not open file");
```

a) The diagnostic output is directly displayed in the output

b) The diagnostic output is pipelined to the output file

c) The line which caused error is compiled again

d) The program is immediately aborted

[View Answer](#)

Answer: a

Explanation: None.

5. Which of the following function can be used to terminate the main() function from another function safely?

a) return(expr);

b) exit(expr);

c) abort();

d) both exit(expr); and abort();

[View Answer](#)

Answer: b

Explanation: None.

6. Which of the following causes an error?

a) Trying to read a file that doesn't exist

b) Inability to write data in a file

c) Failure to allocate memory with the help of malloc

d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

7. What is the purpose of the C function?

```
int ferror(FILE *fp)
```

a) They check for input errors

b) They check for output errors

c) They check for all types of errors

d) They check for error in accessing the file

[View Answer](#)

Answer: b

Explanation: None.

Line Input & Output – 1

1. The syntax of fgets is char *fgets(char *line, int maxline, FILE *fp). Which is true for fgets?

- a) Returns line on success
- b) On end of file or error it returns NULL
- c) Nothing
- d) Both returns line on success & On end of file or error it returns NULL

View Answer

Answer: d

Explanation: None.

2. fputs() function writes a string to a file that only ends with a newline.

- a) True
- b) False
- c) Depends on the standard
- d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  int main()
4.  {
5.      char line[3];
6.      fgets(line, 3, stdin);
7.      printf("%d\n", strlen(line));
8.      return 0;
9.  }
```

- a) 3
- b) 1
- c) Any length since line did not end with null character
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <string.h>
```

```

3.   int main()
4.   {
5.       char line[3];
6.       FILE *fp;
7.       fp = fopen("newfile.txt", "r");
8.       while (fgets(line, 3, fp))
9.           fputs(line, stdout);
10.      return 0;
11.  }

```

- a) Compilation error
- b) Infinite loop
- c) Segmentation fault
- d) No. of lines present in file newfile

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code if 2 characters is typed by the user?

```

1.   #include <stdio.h>
2.   #include <string.h>
3.   int main()
4.   {
5.       char line[3];
6.       fgets(line, 3, stdin);
7.       printf("%d\n", line[2]);
8.       return 0;
9.   }

```

- a) Compilation error
- b) Undefined behaviour
- c) 0
- d) 10(ascii value of newline character)

View Answer

Answer: c

Explanation: None.

6. fputs() adds newline character.

- a) True

- b) False
- c) Depends on the standard
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

7. puts() function adds newline character.

- a) True
- b) False
- c) Depends on the standard
- d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

8. gets() function checks overflow run.

- a) True
- b) False
- c) Depends on the standard
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

9. puts() does the following when it writes to stdout.

- a) Deletes everything
- b) Adds 't' to the line written
- c) Deletes the terminating 'n'
- d) Adds 'n' to the line written

View Answer

Answer: d

Explanation: None.

Line Input & Output – 2

1. What is the size of array "line" used in fgets(line, maxline, *fp) function?

- a) maxline – 1
- b) maxline
- c) maxline + 1
- d) Size is dynamic

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C function when EOF returns?

```
int fputs(char *line, FILE *fp)
```


- a) '◆' character of array line is encountered
- b) 'n' character in array line is encountered
- c) 't' character in array line is encountered
- d) When an error occurs

View Answer

Answer: d

Explanation: None.

3. Identify X library function for line input and output in the following C code?

```

1.    #include <stdio.h>
2.    int X(char *s, FILE *iop)
3.    {
4.        int c;
5.        while (c = *s++)
6.            putc(c, iop);
7.        return ferror(iop) ? EOF : 0;
8.    }
```

- a) getc
- b) putc
- c) fgetc
- d) fputc

View Answer

Answer: d

Explanation: None.

4. Which function has a return type as char pointer?

- a) getline
- b) fputc
- c) fgetc
- d) all of the mentioned

View Answer

Answer: c

Explanation: None.

5. Which of the following is the right declaration for fgetc() inside the library?

- a) int *fgetc(char *line, int maxline, FILE *fp);
- b) char *fgetc(char *line, int maxline, FILE *fp);
- c) char *fgetc(char *line, FILE *fp);
- d) int *fgetc(char *line, FILE *fp);

View Answer

Answer: b

Explanation: None.

6. what is the return value of fputs()?

- a) EOF if an error occurs
- b) Non-negative if no error
- c) EOF if an error occurs & Non-negative if no error
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

7. gets() and puts() operate on _____

- a) stdin and stdout
- b) files
- c) stderr
- d) nothing

View Answer

Answer: a

Explanation: None.

8. gets() does the following when it reads from stdin.

- a) Deletes the 't'
- b) Puts adds it.
- c) Deletes the terminating 'n'
- d) Nothing

View Answer

Answer: c

Explanation: None.

String Operations – 1

1. What will be the output of the following C code?

1. `#include <stdio.h>`
2. `#include <string.h>`
3. `int main()`
4. `{`
5. `char *str = "hello, world";`
6. `char *str1 = "hello, world";`
7. `if (strcmp(str, str1))`
8. `printf("equal");`
9. `else`
10. `printf("unequal");`
11. `}`

- a) equal
- b) unequal
- c) Compilation error
- d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *str = "hello, world";
5.      char str1[15] = "hello wo 9";
6.      strcpy(str, str1);
7.      printf("%s", str1);
8.  }
```

- a) Compilation error
- b) Segmentation Fault
- c) hello, world
- d) hello, wo 9

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  int main()
4.  {
5.      char *str = "hello, world";
6.      char str1[9];
7.      strncpy(str1, str, 9);
8.      printf("%s %d", str1, strlen(str1));
9.  }
```

- a) hello, world 11
- b) hello, wor 9

c) Undefined behaviour

d) Compilation error

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *str = "hello, world\n";
5.      printf("%d", strlen(str));
6.
7.  }
```

a) Compilation error

b) Undefined behaviour

c) 13

d) 11

View Answer

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char str[11] = "hello";
5.      char *str1 = "world";
6.      strcat(str, str1);
7.      printf("%s %d", str, str[10]);
8.  }
```

a) helloworld 0

b) helloworld anyvalue

c) worldhello 0

d) Segmentation fault/code crash

View Answer

Answer: a

Explanation: None.

6. Strcat() function adds null character.

- a) Only if there is space
- b) Always
- c) Depends on the standard
- d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char str[10] = "hello";
5.      char *str1 = "world";
6.      strncat(str, str1, 9);
7.      printf("%s", str);
8.  }
```

- a) helloworld
- b) Undefined behaviour
- c) helloworl
- d) hellowor

View Answer

Answer: a

Explanation: None.

String Operations – 2

1. What type of return-type used in String operations?

- a) void only
- b) void and (char *) only
- c) void and int only
- d) void, int and (char *) only

View Answer

Answer: d

Explanation: None.

2. String operation such as strcat(s, t), strcmp(s, t), strcpy(s, t) and strlen(s) heavily rely upon.

- a) Presence of NULL character
- b) Presence of new-line character
- c) Presence of any escape sequence
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

3. Which pre-defined function returns a pointer to the last occurrence of a character in a string?

- a) strchr(s, c);
- b) strrchr(s, c);
- c) strlchr(s, c);
- d) strfchr(s, c);

View Answer

Answer: b

Explanation: None.

4. Which of the following function compares 2 strings with case-insensitively?

- a) strcmp(s, t)
- b) strcasecmp(s, t)
- c) strcasecmp(s, t)
- d) strchr(s, t)

View Answer

Answer: c

Explanation: None.

5. What will be the value of var for the following C statement?

```
var = strcmp("Hello", "World");
```

- a) -1
- b) 0
- c) 1
- d) strcmp has void return-type

View Answer

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char str[10] = "hello";
5.      char *p = strrchr(str, 'l');
6.      printf("%c\n", *(++p));
7.  }
```

- a) l
- b) o
- c) e

d) Compilation error

[View Answer](#)

Answer: b

Explanation: None.

Character Class Testing & Conversions – 1

1. Which of the following library function is not case-sensitive?

- a) toupper()
- b) tolower()
- c) isdigit()
- d) all of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

2. The following C expression can be substituted for?

if (isalpha(c) && isdigit(c))

- a) if (isalnum(c))
- b) if (isalphanum(c))
- c) if (isalphanumeric(c))
- d) none of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

3. Which of the following will return a non-zero value when checked with isspace(c)?

- a) blank
- b) newline
- c) return
- d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

4. What will be the output of the following C code?

1. `#include <stdio.h>`
2. `#include <ctype.h>`
3. `int main()`
4. `{`
5. `char i = 9;`
6. `if (isdigit(i))`
7. `printf("digit\n");`

```
8.     else
9.         printf("not digit\n");
10.    return 0;
11. }
```

- a) digit
- b) not digit
- c) Depends on the compiler
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <ctype.h>
3.  int main()
4.  {
5.      int i = 9;
6.      if (isdigit(i))
7.          printf("digit\n");
8.      else
9.          printf("not digit\n");
10.     return 0;
11. }
```

- a) digit
- b) not digit
- c) Depends on the compiler
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
```



```

4.     char i = '9';
5.     if (isdigit(i))
6.         printf("digit\n");
7.     else
8.         printf("not digit\n");
9.     return 0;
10.    }

```

- a) digit
 - b) not digit
 - c) Depends on the compiler
 - d) None of the mentioned
- [View Answer](#)

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

1.     #include <stdio.h>
2.     #include <ctype.h>
3.     int main()
4.     {
5.         int i = 0;
6.         if (isspace(i))
7.             printf("space\n");
8.         else
9.             printf("not space\n");
10.        return 0;
11.    }

```

- a) Compile time error
 - b) space
 - c) not space
 - d) None of the mentioned
- [View Answer](#)

Answer: c

Explanation: The value of variable i is 0 which is the NULL character in ASCII. Hence, the output will be printed as "not space".

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  #include <ctype.h>
3.  int main()
4.  {
5.      int i = 32;
6.      if (isspace(i))
7.          printf("space\n");
8.      else
9.          printf("not space\n");
10.     return 0;
11. }

```

- a) Compile time error
- b) space
- c) not space
- d) None of the mentioned

View Answer

Answer: b

Explanation: The ASCII value of space character is 32. Since the variable i stores 32, the output will be printed as "space".

Character Class Testing & Conversions – 2

1. Which is true about isalpha(c), where c is an int that can be represented as an unsigned?

char or EOF.isalpha(c) returns

- a) Non-zero if c is alphabetic
- b) 0 if c is not alphabetic
- c) Both Non-zero if c is alphabetic & 0 if c is not alphabetic
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

2. Which is true about isupper(c), where c is an int that can be represented as an unsigned?

char or EOF.isupper(c) returns

- a) Non-zero if c is upper case
- b) 0 if c is not upper case
- c) Nothing
- d) Both Non-zero if c is upper case & 0 if c is not upper case

View Answer

Answer: d

Explanation: None.

3. Which is true about `isalnum(c)`, where `c` is an int that can be represented as an unsigned?

`char` or `EOF`. `isalnum(c)` returns

- a) Non-zero if `isalpha(c)` or `isdigit(c)`
- b) 0 if not `isalpha(c)` or not `isdigit(c)`
- c) Both Non-zero if `isalpha(c)` or `isdigit(c)` & 0 if not `isalpha(c)` or not `isdigit(c)`
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <ctype.h>
3.  int main()
4.  {
5.      char c = 't';
6.      printf("%d\n", isspace(c));
7.  }
```

- a) Non-zero number
- b) Nothing
- c) Error
- d) t

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <ctype.h>
3.  int main()
4.  {
5.      char c = 't';
6.      printf("is :%c\n", tolower('A'));
7.  }
```

- a) A
- b) a
- c) Non-zero number
- d) Zero

View Answer

Answer: b

Explanation: None.

6. Which types of input are accepted in toupper(c)?

- a) char
- b) char *
- c) float
- d) Both char and char *

View Answer

Answer: a

Explanation: None.

7. What is the difference in the ASCII value of capital and non-capital of the same letter is?

- a) 1
- b) 16
- c) 32
- d) Depends with compiler

View Answer

Answer: c

Explanation: None.

Ungetc – 1

1. ungetc() can be used only with getc().

- a) true
- b) false
- c) depends on the standard
- d) depends on the platform

View Answer

Answer: b

Explanation: None.

2. Which character of pushback is guaranteed per file?

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the platform

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int n;
5.      scanf("%d", &n);
6.      ungetc(n, stdin);
7.      scanf("%d", &n);
8.      printf("%d\n", n);
9.      return 0;
10. }
```

- a) Compile time error
- b) Whatever is typed by the user first time
- c) Whatever is typed by the user second time
- d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      char n[20];
5.      fgets(n, 19, stdin);
6.      ungetc(n[0], stdin);
7.      scanf("%s", n);
8.      printf("%s\n", n);
9.      return 0;
10. }
```

- a) Compile time error
- b) Whatever string user types second time
- c) Whatever string user types first time
- d) First character of whatever user types first time and whatever user types second time

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code considering user typed jkl?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char n[20];
5.      fgets(n, 19, stdin);
6.      ungetc(n[0], stdin);
7.      printf("%s\n", n);
8.      return 0;
9.  }
```

- a) jkl
- b) kl
- c) Undefined behaviour
- d) jk

View Answer

Answer: a

Explanation: None.

6. How many characters for pushback is guaranteed per file while using ungetc(c, fp);?

- a) Only 1 character
- b) Characters within 1 word
- c) Characters within 1st new-line
- d) All characters upto NULL character

View Answer

Answer: a

Explanation: None.

7. Which of the following is the correct syntax for calling function ungetc?

Assume int c and FILE *fp

- a) ungetc(c, *fp);
- b) ungetc(c, fp);
- c) ungetc(fp, c);
- d) ungetc(*fp, c);

View Answer

Answer: b

Explanation: None.

8. ungetc() is used _____

- a) to get a char
- b) to get an int
- c) to push a character back to file
- d) nothing

View Answer

Answer: c

Explanation: None.

Ungetc – 2

1. Which of the following is the correct declaration for ungetc?

- a) int ungetc(int c, FILE fp);
- b) int ungetc(int *c, FILE fp);
- c) int ungetc(int c, FILE *fp);
- d) int ungetc(int *c, FILE *fp);

View Answer

Answer: c

Explanation: None.

2. Which of the following cannot be used with ungetc()?

- a) scanf
- b) getc
- c) getchar
- d) printf

View Answer

Answer: d

Explanation: None.

3. What does the ungetc function return for the following C expression?

`ungetc(c, fp);` //where declarations are `int c` and `FILE *fp`

- a) It returns character c
- b) It returns EOF for an error
- c) Both returns character c and returns EOF for an error
- d) Either returns character c or returns EOF for an error

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C statement?

`int ungetc(int c, FILE *fp)`

- a) Either c or EOF for an error
- b) Nothing
- c) fp

d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

5. Only _____ character of pushback is guaranteed per file when ungetc is used.

a) Two

b) One

c) Many

d) Zero

[View Answer](#)

Answer: b

Explanation: None.

6. ungetc() may be used with _____

a) scanf

b) getc

c) getchar

d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

7. What is the syntax of ungetc()?

a) void ungetc(int c, FILE *fp)

b) int ungetc(int c, FILE *fp)

c) int ungetc(String c, FILE *fp)

d) int getc(int c, FILE *fp)

[View Answer](#)

Answer: b

Explanation: None.

Storage Management – 1

1. The function _____ obtains a block of memory dynamically.

a) calloc

b) malloc

c) both calloc & malloc

d) free

[View Answer](#)

Answer: c

Explanation: None.

2. void * malloc(size_t n) returns?

a) Pointer to n bytes of uninitialized storage

b) NULL if the request can be satisfied

c) Nothing

d) None of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

3. calloc() returns storage that is initialized to.

a) Zero

b) Null

c) Nothing

d) One

[View Answer](#)

Answer: a

Explanation: None.

4. In function free(p), p is a _____

a) int

b) pointer returned by malloc()

c) pointer returned by calloc()

d) pointer returned by malloc() & calloc()

[View Answer](#)

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      char *p = calloc(100, 1);
5.      p = "welcome";
6.      printf("%s\n", p);
7.  }
```

a) Segmentation fault

b) Garbage

c) Error

d) welcome

[View Answer](#)

Answer: d

Explanation: None.

6. Memory allocation using malloc() is done in _____

a) Static area

b) Stack area

c) Heap area

d) Both Stack & Heap area

[View Answer](#)

Answer: c

Explanation: None.

7. Why do we write (int *) before malloc?

```
int *ip = (int *)malloc(sizeof(int));
```

a) It is for the syntax correctness

b) It is for the type-casting

c) It is to inform malloc function about the data-type expected

d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

8. Which of the following is used during memory deallocation in C?

a) remove(p);

b) delete(p);

c) free(p);

d) terminate(p);

[View Answer](#)

Answer: c

Explanation: None.

Storage Management – 2

1. Which of the following will return a result most quickly for searching a given key?

a) Unsorted Array

b) Sorted Array

c) Sorted linked list

d) Binary Search Tree

[View Answer](#)

Answer: d

Explanation: None.

2. On freeing a dynamic memory, if the pointer value is not modified, then the pointer points to.

a) NULL

b) Other dynamically allocated memory

c) The same deallocated memory location

d) It points back to the location it was initialized with

[View Answer](#)

Answer: c

Explanation: None.

3. Which of the following should be used for freeing the memory allocated in the following C code?

```
1.    #include <stdio.h>
```

```

2.   struct p
3.   {
4.       struct p *next;
5.       int x;
6.   };
7.   int main()
8.   {
9.       struct p *p1 = (struct p*)malloc(sizeof(struct p));
10.      p1->x = 1;
11.      p1->next = (struct p*)malloc(sizeof(struct p));
12.      return 0;
13.  }

```

a)

```

free(p1);
free(p1->next)

```

b)

```

free(p1->next);
free(p1);

```

c) free(p1);

d) all of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```

1.   #include <stdio.h>
2.   struct p
3.   {
4.       struct p *next;
5.       int x;
6.   };
7.   int main()
8.   {

```

```

9.    struct p *p1 = calloc(1, sizeof(struct p));
10.   p1->x = 1;
11.   p1->next = calloc(1, sizeof(struct p));
12.   printf("%d\n", p1->next->x);
13.   return 0;
14.  }

```

- a) Compile time error
- b) 1
- c) Somegarbage value
- d) 0

View Answer

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    struct p
3.    {
4.        struct p *next;
5.        int x;
6.    };
7.    int main()
8.    {
9.        struct p* p1 = malloc(sizeof(struct p));
10.       p1->x = 1;
11.       p1->next = malloc(sizeof(struct p));
12.       printf("%d\n", p1->next->x);
13.       return 0;
14.    }

```

- a) Compile time error
- b) 1
- c) Somegarbage value
- d) 0

View Answer

Answer: c

Explanation: None.

6. calloc() initialize memory with all bits set to zero.

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

View Answer

Answer: a

Explanation: None.

7. What if size is zero in the following C statement?

realloc(ptr, size)

- a) Allocate a memory location with zero length
- b) Free the memory pointed to by ptr
- c) Undefined behaviour
- d) Doesn't do any reallocation of ptr i.e. no operation

View Answer

Answer: b

Explanation: None.

Mathematical Functions – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <math.h>
3.  int main()
4.  {
5.      int i = 90;
6.      printf("%f\n", sin(i));
7.      return 0;
8.  }
```

- a) Compile time error
- b) Undefined behaviour
- c) 0.893997
- d) 1.000000

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  #include <math.h>
3.  int main()
4.  {
5.      unsigned int i = -1;
6.      printf("%f\n", fabs(i));
7.      return 0;
8.  }

```

- a) Compile time error
- b) 1
- c) -1
- d) None of the mentioned

View Answer

Answer: d

Explanation: None.

3. function fabs defined math.h header file takes the argument of type integer.

- a) True
- b) False
- c) Depends on the implementation
- d) Depends on the standard

View Answer

Answer: b

Explanation: None.

4. log(x) function defined in math.h header file is _____

- a) Natural base logarithm
- b) Logarithm to the base 2
- c) Logarithm to the base 10
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  #include <math.h>
3.  int main()
4.  {
5.      int i = 10;

```

```
6.    printf("%f\n", log10(i));  
7.    return 0;  
8.    }
```

- a) Compile time error
- b) 1.000000
- c) 2.302585
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

6. What type of inputs are accepted by mathematical functions?

- a) short
- b) int
- c) float
- d) double

[View Answer](#)

Answer: d

Explanation: None.

7. In linux, apart from including math header file, the program is successfully executed by which of the following?

- a) cc filename.c
- b) cc filename.c -lc
- c) cc -math filename.c
- d) cc -lm filename.c

[View Answer](#)

Answer: d

Explanation: None.

8. Which of the following is not a valid mathematical function?

- a) frexp(x);
- b) atan2(x,y);
- c) srand(x);
- d) fmod(x);

[View Answer](#)

Answer: d

Explanation: None.

Mathematical Functions – 2

1. Which of the following mathematical function requires 2 parameter for successful function call?

- a) fmod();
- b) div();
- c) atan2();

d) all of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

2. Which mathematical function among the following does NOT require int parameters?

a) div(x, y);

b) srand(x);

c) sqrt(x);

d) all of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

3. What will sin(x) returns?

a) sine of x where x is in radians

b) sine of x where x is in degree

c) cosine of x where x is in radians

d) cosine of x where x is in degree

[View Answer](#)

Answer: a

Explanation: None.

4. What will cos(x) return?

a) sine of x where x is in radians

b) sine of x where x is in degree

c) cosine of x where x is in radians

d) cosine of x where x is in degree

[View Answer](#)

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```
2.    #include <math.h>
```

```
3.    void main()
```

```
4.    {
```

```
5.        int k = pow(2, 3);
```

```
6.        printf("%d\n", k);
```

```
7.    }
```

a) 9

b) 8

c) -1

d) 6

[View Answer](#)

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <math.h>
3.  void main()
4.  {
5.      int k = fabs(-87);
6.      printf("%d\n", k);
7.  }
```

a) -87

b) 87

c) 78

d) error

[View Answer](#)

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <math.h>
3.  void main()
4.  {
5.      int k = sqrt(-4);
6.      printf("%d\n", k);
7.  }
```

a) -2

b) 2

c) Compile time error

d) NaN

[View Answer](#)

Answer: d

Explanation: None.

8. Which among the following mathematical function do not have a “double” return-type?

a) srand(x);

- b) ceil(x);
- c) floor(x);
- d) both ceil(x); and floor(x);

View Answer

Answer: a

Explanation: None.

Random Number Generation – 1

1. What is function srand(unsigned)?

- a) Sets the seed for rand
- b) Doesn't exist
- c) Is an error
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

2. Which is the best way to generate numbers between 0 to 99?

- a) rand()-100
- b) rand()%100
- c) rand(100)
- d) srand(100)

View Answer

Answer: b

Explanation: None.

3. Which is the correct way to generate numbers between minimum and maximum(inclusive)?

- a) minimum + (rand() % (maximum – minimum));
- b) minimum + (rand() % (maximum – minimum + 1));
- c) minimum * (rand() % (maximum – minimum))
- d) minimum – (rand() % (maximum + minimum));

View Answer

Answer: b

Explanation: None.

4. rand() and srand() functions are used _____

- a) To find sqrt
- b) For and operations
- c) For or operations
- d) To generate random numbers

View Answer

Answer: d

Explanation: None.

5. What is the return type of rand() function?

- a) short

- b) int
- c) long
- d) double

View Answer

Answer: b

Explanation: None.

6. Which of the following can be used for random number generation?

- a) random()
- b) rnd()
- c) rndm()
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

7. Which of the following snippet will effectively generate random numbers?

- a) rand();
- b) rand(10);
- c) rand(time(NULL));
- d) all of the mentioned

View Answer

Answer: a

Explanation: None.

8. Which among the following is correct function call for rand() and random()?

- a) rand() and random();
- b) rand() and random(1);
- c) rand(1) and random(1);
- d) rand(1) and random();

View Answer

Answer: a

Explanation: None.

9. For the function call time(), what type of parameter is accepted?

- a) int
- b) int *
- c) time_t
- d) time_t *

View Answer

Answer: d

Explanation: None.

Random Number Generation – 2

1. What will be the output of the following C code?

```
1.    #include <stdio.h>
```

```

2.    #include <stdlib.h>
3.    int main()
4.    {
5.        printf("%d\n", rand() % 1000);
6.        return 0;
7.    }

```

- a) Compile time error
- b) An integer between 0-1000
- c) An integer between 0-999 including 0 and 999
- d) An integer between 0-1000 including 1000

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    #include <stdlib.h>
3.    int main()
4.    {
5.        srand(9000);
6.        printf("%d\n", rand());
7.        return 0;
8.    }

```

- a) Compile time error
- b) An integer in the range 0 to RAND_MAX
- c) A double in the range 0 to 1
- d) A float in the range 0 to 1

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```

1.    #include <stdio.h>
2.    int main()
3.    {
4.        printf("%d\n", srand(9000));

```

5. return 0;

6. }

- a) Compile time error
- b) An integer in the range 0 to 9000
- c) A float in the range 0 to 1
- d) A double in the range 0 to 9000

View Answer

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.       #include <stdio.h>
2.       int main()
3.       {
4.        srand(time(NULL));
5.        printf("%d\n", rand());
6.        return 0;
7.       }
```

- a) Compile time error
- b) An integer in the range 0 to RAND_MAX
- c) A double in the range 0 to 1
- d) A float in the range 0 to 1

View Answer

Answer: b

Explanation: None.

5. In the below C program, every time program is run different numbers are generated.

```
1.       #include <stdio.h>
2.       #include <stdlib.h>
3.       int main()
4.       {
5.        printf("%d\n", rand());
6.        return 0;
7.       }
```

- a) True
- b) False
- c) Depends on the platform

d) Depends on the compiler

[View Answer](#)

Answer: b

Explanation: None.

6. In the following C program, every time program is run different numbers are generated.

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        srand(time(NULL));
5.        printf("%d\n", rand());
6.        return 0;
7.    }
```

a) True

b) False

c) Depends on the platform

d) Depends on the compiler

[View Answer](#)

Answer: a

Explanation: None.

7. Which of these is a correct way to generate numbers between 0 to 1(inclusive) randomly?

a) rand() / RAND_MAX

b) rand() % 2

c) rand(0, 1)

d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

C printf – 1

1. The syntax of printf() function is printf("control string", variable list) ;what is the prototype of the control string?

a) %[flags][.precision][width][length]specifier

b) %[flags][length][width][.precision]specifier

c) %[flags][width][.precision][length]specifier

d) %[flags][.precision][length][width]specifier

[view Answer](#)

Answer: c

Explanation: The prototype of control string is %[flags][width][.precision][length]specifier. Each control string must begin with % sign.

2. The parameter control string in the printf () is a C String that contains text to be _____

- a) taken from a standard output device
- b) written on to the standard output device
- c) received from the standard output device
- d) nothing can be said

view Answer

Answer: b

Explanation: After the control string, the function can have many additional arguments as specified in the control string, this parameter contains the text to be written on to the standard output device.

3. Output justification such as decimal point, numerical sign, trailing zeros or octal are specified.

- a) specifier
- b) flags
- c) precision
- d) decimal

view Answer

Answer: b

Explanation: Flags specify output justification such as Left-justify within the data given field width, Displays the data with its numeric sign, used to provide additional specifiers like o, x, X for octal, left padding of a number.

4. What symbol is used to Left-justify within the data given field width?

- a) -(minus sign)
- b) +(plus sign)
- c) #
- d) 0

view Answer

Answer: a

Explanation: To left-justify the data use minus sign(-) in the flags field.

5. What specifies the minimum number of characters to print after being padded with zeros or blank spaces?

- a) flags
- b) length
- c) width
- d) precision

view Answer

Answer: c

Explanation: width specifies the minimum number of positions in the output.

6. The maximum number of characters to be printed is specified by _____

- a) precision
- b) width
- c) length
- d) flags

view Answer

Answer: a

Explanation: Precision specifies the maximum number of characters to print.

7. _____ is used to define the type and the interpretation of the value of the corresponding argument.

- a) precision
- b) specifiers
- c) flags
- d) decimal

view Answer

Answer: b

Explanation: Specifiers is used to define the type and the interpretation of the value of the corresponding argument. Example: c for a single character, d for decimal values etc.

8. A conversion specification %7.4f means _____

- a) print a floating point value of maximum 7 digits where 4 digits are allotted for the digits after the decimal point
- b) print a floating point value of maximum 4 digits where 7 digits are allotted for the digits after the decimal point
- c) print a floating point value of maximum 7 digits
- d) print a floating point value of minimum 7 digits where 4 digits are allotted for the digits after the decimal point

view Answer

Answer: a

Explanation: The conversion specification %7.4f means that it will print floating point number maximum of 7 digits and 4 digits after the decimal point.

9. Choose the correct description for control string %-+7.2f.

- a) – means display the sign, + means left justify, 7 specifies the width and 2 specifies the precision
- b) – means left justify, + means display the sign, 7 specifies the width and 2 specifies the precision
- c) – means display the sign, + means left justify, 7 specifies the precision and 2 specifies the width
- d) – means left justify, + means display the sign, 7 specifies the precision and 2 specifies the width

view Answer

Answer: b

Explanation: The given control string %-+7.2f means that – is for left justify, + to display sign, 7 specifies the precision and 2 specifies the width.

10. What error is generated on placing an address operator with a variable in the printf statement?

- a) compile error
- b) run-time error
- c) logical error
- d) no error

view Answer

Answer: b

Explanation: Placing an address operator with a variable in the printf statement will generate a run-time error.

C printf – 2

1. If by mistake you specify more number of arguments, the excess arguments will _____

- a) be ignored
- b) produce compile error
- c) produce run-time error
- d) produce logical error

[View Answer](#)

Answer: a

Explanation: The excess arguments will simply be ignored.

2. What happens when zero flag is used with left justification?

- a) data is padded with zeros
- b) zero flag is ignored
- c) data is padded with blank spaces
- d) will give error

[View Answer](#)

Answer: b

Explanation: Zero flag is not considered when used with left justification because adding zeros after a number changes its value.

3. For floating point numbers, the precision flag specifies the number of decimal places to be printed.

When no precision modifier is specified, printf() prints _____

- a) six decimal positions
- b) five decimal positions
- c) four decimal positions
- d) three decimal positions

[View Answer](#)

Answer: a

Explanation: Its format can be given as “.m”, where m specifies the number of decimal digits when no precision modifier is specified, printf prints six decimal positions.

4. What will the given code result in `printf("\n you are\"awesome \" ");`?

- a) compile error
- b) run-time error
- c) you are "awesome"
- d) you are awesome

[View Answer](#)

Answer: c

Explanation: The above given code uses `\"<word>\"` to display the word within double inverted commas on standard output screen.

5. What will be the output for the given code `printf("\n The number is %07d",1212);`

- a) The number is 0001212
- b) The number is 1212
- c) The number is 1212

d) The number is 1212000

[View Answer](#)

Answer: a

Explanation: 0 in the above code is Flags. The number is left-padded with zeros(0) instead of spaces.

6. What will be the output of the following code?

```
char t='N';
```

```
printf("\n %c \n %3c \n %5c",t,t,t);
```

a) N

 N

 N

b) N

 N

 N

c) N

 N

 N

d) N N N

[View Answer](#)

Answer: b

Explanation: In the given code each argument is printed on a new line due to control character \n. Width mentioned in the above code is 1,3,5 hence the character is printed on a new line after being padded with blank spaces.

7. Select the right explanation to the given code.

```
printf("%*. *f", 5,4,5700);
```

a) the minimum field width has to be 4, the precision is given to be 5, and the value to be displayed is 5700

b) the minimum field width is 5, the precision is 4, and the value to be displayed is 5700

c) compile error

d) run-time error

[View Answer](#)

Answer: b

Explanation: The minimum field width and precision specifiers are usually constants. They can also be provided by arguments to printf(). This is done by using * modifier as shown in the given code.

8. What will be the output of the following C code?

```
char str[] = "Hello Nancy";
```

```
printf("\n %.7s", str);
```

a) Hello Nan

b) Hello

c) Hello N

d) Hello Nancy

[View Answer](#)

Answer: c

Explanation: The output for the code must be 7 characters including white spaces.

9. What will be the output of the following C code?

```
char str[] = "Too Good";
```

```
printf("\n %7s",str);
```

a) Too Good

b) Too G

c) Too Go

d) Too

[View Answer](#)

Answer: a

Explanation: The complete string "Too Good" is printed. This is because if data needs more space than specified, then printf overrides the width specified by the user.

10. What will be the output of the following C code?

```
printf("\n Output: %5d \t %x \t %#x", 234,234,234);
```

a) Output:234EA0xEA

b) Output:00234 EA 0xEA

c) Output: 234 EA 0xEA

d) ERROR

[View Answer](#)

Answer: c

Explanation: The control character \t is used to provide gap between the words. %5d – the width of the string is set to 5, characters are printed after being padded with blank spaces.%x, %#x is additional specifiers for octal and hexadecimal values.

C scanf – 1

1. The syntax of the scanf() is scanf("control string ", arg1,arg2,arg3,...,argn); the prototype of control string is _____

a) [=][width][modifiers]type=]

b) [=][modifiers][width]type=]

c) [=][width] [modifiers]]

d) [width][modifiers]

[View Answer](#)

Answer: a

Explanation: scanf() starts with the symbol % followed by the width, modifier, type of the argument.

2. What is the use of symbol * in the control string as shown [=][*][width] [modifiers] type=]?

a) * is optional and used when the data should be read from the stream but ignored

b) * is not optional, used to read data from the stream but it is not ignored

c) * is not optional, it is used to read data stream but ignored

d) * is optional and used to read data from stream but it is not ignored

[View Answer](#)

Answer: a

Explanation: * is an optional argument, it indicates that data should be read from the stream but ignored (not stored in a memory location)

3. What action is carried out by scanf if a user enters any blank spaces, tabs, and newlines?

- a) consider as input
- b) ignores it
- c) produces error
- d) nothing can be said

[View Answer](#)

Answer: b

Explanation: The scanf() function ignores any blank spaces, tabs, and newlines entered by the user. This scanf() function just returns the number of input fields successfully scanned and stored.

4. What error will generate if the read and write parameters are not separated by commas?

- a) run-time error
- b) compile error
- c) logical error
- d) no error

[View Answer](#)

Answer: b

Explanation: A compile error will be generated if the read and write parameters are not separated by commas.

5. What will be the output of the following C code?

```
char str[] = "Good";
```

```
scanf("%s", str);
```

- a) compile error
- b) run-time error
- c) good
- d) logical error

[View Answer](#)

Answer: c

Explanation: String can be read from the stream without the use of address of operator (&).

6. What will be the output of the following C code?

```
scanf("%d %d %d",&n1,&n2);
```

- a) read data for two
- b) generate error
- c) read data for three
- d) nothing can be said

[View Answer](#)

Answer: b

Explanation: The following scanf() statement will generate an error as no variable address is given for the third conversion specification.

7. What form the data must be entered for the given C code?

```
scanf("%d / %d", &n1,&n2);
```

- a) 6 9
- b) 6/9
- c) compile error
- d) run-time error

View Answer

Answer: b

Explanation: The slash in the control String are neither white space characters nor a part of conversion specification, so the user must enter data of the form 6/9.

8. A fatal error will be generated if the format string is ended with a white space character.

- a) true
- b) false

View Answer

Answer: a

Explanation: An error will be generated if the format string %s is ended with white space character.

9. Explain the format string "%5d%s %c"

- a) five characters as a decimal integer, then reads the remaining as a string and then scans the first non-whitespace character
- b) compile error
- c) run-time error
- d) read first five characters as a decimal and ignore the rest

View Answer

Answer: a

Explanation: The above format string reads the first five characters as a decimal integer, then reads the remaining as a string until a space, newline or tab is found, then reads the first non-whitespace character.

10. _____ is an optional argument that gives the maximum number of characters to be read.

- a) modifiers
- b) width
- c) precision
- d) length

View Answer

Answer: b

Explanation: Width is the argument that gives the maximum number of characters to be read. Few characters will be read if the scanf function encounters white space and it will stop processing further.

C scanf – 2

1. Select the correct value of i from given options `i=scanf("%d %d", &a, &b);`

- a) 1
- b) 2
- c) 3
- d) No value assigned

View Answer

Answer: b

Explanation: i stores the number of read data from the stream. It is useful for detecting an error in data input.

2. If the user enters 1 3.2 s, what value will be returned by the `scanf()`?

`scanf("%d %f %c", &s1, &s2, &s3);`

- a) 1
- b) 2
- c) 3
- d) No return value

View Answer

Answer: c

Explanation: When the `scanf()` function completes reading all the data values, it returns number of values that are successfully read.

3. If the user enters 1 s 3.2, what value will be returned by the `scanf()`?

`scanf("%d %f %c", &a, &b, &c);`

- a) 1
- b) 2
- c) 3
- d) no return value

View Answer

Answer: a

Explanation: `scanf()` returns the number of values that are successfully read. In the above statement, only integer value is read successfully.

4. What error will be generated on using incorrect specifier for the datatype being read?

- a) compile error
- b) run-time error
- c) logical error
- d) no error

View Answer

Answer: b

Explanation: Using an incorrect specifier for the datatype being read will generate a run-time error.

5. What is the prototype of `scanf` function?

- a) `scanf("controlstring", arg1, arg2, arg3, ..., argn);`

- b) `scanf("control string", variable list);`
- c) `scanf(" variable list", control string);`
- d) `scanf("arg1,arg2,arg3,...,argn", control string);`

View Answer

Answer: a

Explanation: The syntax of the `scanf()` can be given as, `scanf("control string", arg1,arg2,arg3,...,argn);`

6. Control string specifies the type and format of the data that has to be obtained from the keyboard.

- a) true
- b) false

View Answer

Answer: a

Explanation: The control string specifies the type and format of the data that has to be obtained from the keyboard and store in the memory locations pointed by the arguments `arg1,arg2....argn`.

7. What is the qualifying input for the type specifier G?

- a) floating point numbers
- b) floating point numbers in exponential format
- c) floating point numbers in the shorter of exponential format
- d) not a type specifier

View Answer

Answer: c

Explanation: G is a type specifier used to take an input of floating point numbers in the shorter of exponential format.

8. `scanf()` is a predefined function in _____ header file.

- a) `stdlib. h`
- b) `ctype. h`
- c) `stdio. h`
- d) `stdarg. h`

View Answer

Answer: c

Explanation: `scanf()` is a predefined function in "`stdio.h`" header file. `printf` and `scanf()` carry out input and output functions in C. These functions statements are present in the header file `stdio.h`.

9. What does the C statement given below says?

```
scanf("%7s",ch);
```

- a) read string with minimum 7 characters.
- b) read string with maximum 7 characters
- c) read string exactly to 7 characters
- d) read string with any number of characters

View Answer

Answer: b

Explanation: In the above statement the control string specifies the size of string to be 7(i.e only 7 characters can be entered in a string).

10. What is the meaning of the following C statement?

```
scanf("%[^\n]s", ch);
```

- a) read all character except new line
- b) read all characters
- c) read only new line character
- d) syntax error

View Answer

Answer: a

Explanation: The symbol ^ when used before a escape sequence, does not read from the console.

File Operations – 1

1. Which one of the following is correct syntax for opening a file.

- a) FILE *fopen(const *filename, const char *mode)
- b) FILE *fopen(const *filename)
- c) FILE *open(const *filename, const char *mode)
- d) FILE open(const*filename)

View Answer

Answer: a

Explanation: fopen() opens the named file, and returns a stream, or NULL if the attempt fails.

2. What is the function of the mode 'w+'?

- a) create text file for writing, discard previous contents if any
- b) create text file for update, discard previous contents if any
- c) create text file for writing, do not discard previous contents if any
- d) create text file for update, do not discard previous contents if any

View Answer

Answer: b

Explanation: w+ is a mode used to open a text file for update (i. e., writing and reading), discard previous contents if any.

3. If the mode includes b after the initial letter, what does it indicate?

- a) text file
- b) big text file
- c) binary file
- d) blueprint text

View Answer

Answer: c

Explanation: If the mode consists of letter b after the first letter as in, "rb" or "w+b", it indicates binary file.

4. fflush(NULL) flushes all _____

- a) input streams
- b) output streams
- c) previous contents

d) appended text

[View Answer](#)

Answer: b

Explanation: fflush(FILE *stream) – fflush() causes any buffered but unwritten to be written on an Output stream. On an input stream, the effect is undefined. fflush(NULL) flushes all output streams.

5. _____ removes the named file, so that a subsequent attempt to open it will fail.

a) remove(const *filename)

b) remove(filename)

c) remove()

d) fclose(filename)

[View Answer](#)

Answer: a

Explanation: remove(const *filename) removes the named file, so that a subsequent attempt to open it will fail. It returns non-zero if the attempt fails.

6. What is the function of FILE *tmpfile(void)?

a) creates a temporary file of mode "wb+"

b) creates a temporary file of mode "wb"

c) creates a temporary file of mode "w"

d) creates a temporary file of mode "w+"

[View Answer](#)

Answer: a

Explanation: A temporary file is created by tmpfile() function of mode "wb+" that will be automatically removed when closed or when the program terminates normally.

7. What does tmpfile() returns when it could not create the file?

a) stream and NULL

b) only stream

c) only NULL

d) does not return anything

[View Answer](#)

Answer: a

Explanation: tmpfile() returns a stream or NULL if it could not create the file.

8. Choose the right statement for fscanf() and scanf()

a) fscanf() can read from standard input whereas scanf() specifies a stream from which to read

b) fscanf() can specifies a stream from which to read whereas scanf() can read only from standard input

c) fscanf() and scanf() has no difference in their functions

d) fscanf() and scanf() can read from specified stream

[View Answer](#)

Answer: b

Explanation: The fscanf() is similar to the scanf() function, except that the first argument of fscanf() specifies a stream from which to read whereas scanf() can read from standard input.

9. EOF is an integer type defined in stdio. hand has a value _____

- a) 1
- b) 0
- c) NULL
- d) – 1

View Answer

Answer: d

Explanation: EOF is an integer type defined in stdio. hand has a value – 1.

10. fwrite() can be used only with files that are opened in binary mode.

- a) true
- b) false

View Answer

Answer: a

Explanation: fwrite() can be used to write characters, integers, or structures to a file. However, fwrite() can be used only with files opened in binary mode.

File Operations – 2

1. what is the function of fputs()?

- a) read a line from a file
- b) read a character from a file
- c) write a character to a file
- d) write a line to a file

View Answer

Answer: d

Explanation: The fputs() is used to write a line to a file. fputs() syntax can be written as
`int fputs(const char *str, FILE *stream);`

2. What does the following C code snippet mean?

```
char *gets(char *s)
```

- a) reads the next input line into the array s
- b) writes the line into the array s
- c) reads the next input character into the array s
- d) write a character into the array

View Answer

Answer: a

Explanation: gets() reads the next input line into the array s, terminating newline is replaced with '\0'. It returns s, or NULL if end of file or error occurs.

3. Which function will return the current file position for stream?

- a) fgetpos()
- b) fseek()
- c) ftell()
- d) fsetpos()

View Answer

Answer: c

Explanation: The current file position is returned by `ftell()` function for stream, or `-1L` on error.

4. Select the right explanation for the following C code snippet.

```
int fgetpos(FILE *stream, fpos_t *s)
```

- a) records the current position in stream in *s
- b) sets the file position for stream in *s
- c) positions stream at the position recorded in *s
- d) reads from stream into the array ptr

View Answer

Answer: a

Explanation: `fgetpos()` records the current position in stream in *s, for subsequent use by `fsetpos()`. The type `fpos_t` is suitable for recording such values.

5. Which functions is declared in `<errno.h>`?

- a) `fseek()`
- b) `ftell()`
- c) `ferror()`
- d) `fsetpos()`

View Answer

Answer: c

Explanation: `ferror()` is declared under `<errno.h>`. `ferror()` returns non-zero if the error indicator for stream is set.

6. `setvbuf()` and `setbuf()` function controls buffering for the stream.

- a) true
- b) false

View Answer

Answer: a

Explanation: `setvbuf()` and `setbuf()` controls buffering for the stream. If `buff` is `NULL`, buffering is turned off for the stream.

7. The functions `vprintf()`, `vfprintf()`, and `vsprintf()` are not equivalent to the corresponding `printf()` functions except the variable argument list.

- a) true
- b) false

View Answer

Answer: b

Explanation: The functions `vprintf()`, `vfprintf()`, and `vsprintf()` are similar to the corresponding `printf()` functions except that the variable argument list is replaced by `arg`.

8. The _____ function reads atmost one less than the number of characters specified by size from the given stream and it is stored in the string `str`.

- a) `fgetc()`
- b) `fgets()`
- c) `fputc()`

d) fputs()

[View Answer](#)

Answer: b

Explanation: The fgets() function reads one less than the number of characters indicated by the size from the given stream and it is stored in the string str. The fgets() terminates as soon as it encounters either a newline character, EOF, or other error.

9. What does the following C code snippet mean?

```
int ungetc(int c, FILE *stream)
```

- a) pushes c back onto a stream
- b) deletes c from the stream
- c) reads frequency of c in stream
- d) no action is taken by the command

[View Answer](#)

Answer: a

Explanation: ungetc() pushes c back onto stream, where it will be returned on the next read. Only one character of pushback per stream is Guaranteed.

10. Choose the correct difference between getc() and fgetc().

- a) If it is not a macro, it may evaluate stream more than once
- b) if it is a macro, it may not evaluate stream more than once
- c) if it is a macro, it may evaluate stream more than once
- d) no difference between fgetc() and getc()

[View Answer](#)

Answer: c

Explanation: getc() is equivalent to fgetc() except that if it is a macro, it may evaluate more than once.

MCQs on Floating Point & Sizeof Operator in C

Float Datatype – 1

1. How many digits are present after the decimal in float value?

- a) 1
- b) 3
- c) 6
- d) 16

[View Answer](#)

Answer: c

Explanation: None.

2. Which among the following is never possible as an output for a float?

- a) 3.666666
- b) 3.666
- c) 3
- d) None of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

3. In a 32-bit compiler, which 2 types have the same size?

- a) char and short
- b) short and int
- c) int and float
- d) float and double

[View Answer](#)

Answer: c

Explanation: None.

4. What is the size of float in a 32-bit compiler?

- a) 1
- b) 2
- c) 4
- d) 8

[View Answer](#)

Answer: c

Explanation: None.

5. Loss in precision occurs for typecasting from _____

- a) char to short
- b) float to double
- c) long to float
- d) float to int

[View Answer](#)

Answer: d

Explanation: None.

6. In the following C code, the union size is decided by?

1. union temp
2. {
3. char a;
4. int b;
5. float c;
6. };

- a) char
- b) int
- c) float
- d) both int and float

[View Answer](#)

Answer: d

Explanation: None.

7. %f access specifier is used for _____

- a) Strings
- b) Integral types
- c) Floating type
- d) All of the mentioned

[View Answer](#)

Answer: c

Explanation: None.

8. Select the odd one out with respect to type?

- a) char
- b) int
- c) long
- d) float

[View Answer](#)

Answer: d

Explanation: None.

Float Datatype – 2

1. What will be the output of the following C code?

```
#include <stdio.h>

printf("%.0f", 2.89);
```

- a) 2.890000
- b) 2.89

c) 2

d) 3

[View Answer](#)

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
1.    #include <stdio.h>
2.    int main()
3.    {
4.        float a = 2.4555555555555;
5.        printf("%f", a);
6.    }
```

a) 2.455555

b) 2.455556

c) 2.456

d) 2.46

[View Answer](#)

Answer: a

Explanation: None.

3. Which of the following % operation is invalid?

a) 2 % 4;

b) 2 % 4l;

c) 2 % 4f;

d) Both 2 % 4l; and 2 % 4f;

[View Answer](#)

Answer: c

Explanation: None.

4. Which data type is suitable for storing a number like?

10.0000000001

a) int

b) float

c) double

d) both float and double

[View Answer](#)

Answer: c

Explanation: None.

5. Modulus for float could be achieved by?

a) a % b

- b) modulus(a, b);
- c) fmod(a, b);
- d) mod(a, b);

View Answer

Answer: c

Explanation: None.

6. Predict the data type of the following mathematical operation?

$2 * 9 + 3 / 2 . 0$

- a) int
- b) long
- c) float
- d) double

View Answer

Answer: d

Explanation: None.

7. %lf is used to display?

- a) float
- b) long float
- c) double
- d) all of the mentioned

View Answer

Answer: c

Explanation: None.

Sizeof Keyword – 1

1. What is the sizeof(char) in a 32-bit C compiler?

- a) 1 bit
- b) 2 bits
- c) 1 Byte
- d) 2 Bytes

View Answer

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>

printf("%d", sizeof('a'));
```

- a) 1
- b) 2
- c) 4
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

3. Size of an array can be evaluated by _____

(Assuming array declaration `int a[10];`)

- a) `sizeof(a);`
- b) `sizeof(*a);`
- c) `sizeof(a[10]);`
- d) `10 * sizeof(a);`

[View Answer](#)

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.
3.  union temp
4.  {
5.      char a;
6.      char b;
7.      int c;
8.  }t;
9.  int main()
10. {
11.     printf("%d", sizeof(t));
12.     return 0;
13. }
```

- a) 1
- b) 2
- c) 4
- d) 6

[View Answer](#)

Answer: c

Explanation: None.

5. Which of the following is not an operator in C?

- a) ,
- b) `sizeof()`
- c) ~
- d) None of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

6. Which among the following has the highest precedence?

- a) &
- b) <<
- c) sizeof()
- d) &&

[View Answer](#)

Answer: c

Explanation: None.

7. What is the sizeof(void) in a 32-bit C?

- a) 0
- b) 1
- c) 2
- d) 4

[View Answer](#)

Answer: b

Explanation: None.

8. What type of value does sizeof return?

- a) char
- b) short
- c) unsigned int
- d) long

[View Answer](#)

Answer: c

Explanation: None.

Sizeof Keyword – 2

1. Which among the following is never possible in C when members are different in a structure and union?

//Let P be a structure

//Let Q be a union

- a) sizeof(P) is greater than sizeof(Q)
- b) sizeof(P) is less than sizeof(Q)
- c) sizeof(P) is equal to sizeof(Q)
- d) none of the mentioned

[View Answer](#)

Answer: d

Explanation: None.

2. Which among the following is never possible in C when members in a structure are the same as that in a union?

//Let P be a structure

//Let Q be a union

- a) sizeof(P) is greater than sizeof(Q)
- b) sizeof(P) is equal to sizeof(Q)
- c) sizeof(P) is less than to sizeof(Q)
- d) none of the mentioned

View Answer

Answer: c

Explanation: None.

3. What will be the size of the following C structure?

- 1. #include <stdio.h>
- 2. struct temp
- 3. {
- 4. int a[10];
- 5. char p;
- 6. };

- a) 5
- b) 11
- c) 41
- d) 44

View Answer

Answer: d

Explanation: None.

4. What will be the output of the following C code?

- 1. #include <stdio.h>
- 2. main()
- 3. {
- 4. int a = 1;
- 5. printf("size of a is %d, ", sizeof(++a));
- 6. printf("value of a is %d", a);
- 7. };

- a) size of a is 4, value of a is 1
- b) size of a is 4, value of a is 2
- c) size of a is 2, value of a is 2
- d) size of a is 2, value of a is 2

View Answer

Answer: a

Explanation: None.

5. Which among the following statement is right?

- a) `sizeof(struct stemp*) > sizeof(union utemp*) > sizeof(char *)`
- b) `sizeof(struct stemp*) < sizeof(union utemp*) < sizeof(char *)`
- c) `sizeof(struct stemp*) = sizeof(union utemp*) = sizeof(char *)`
- d) the order Depends on the compiler

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code?

- ```
1. #include <stdio.h>
2. printf("%d", sizeof(strlen("HELLOWORLD")));
```

- a) Output, 4
- b) Output, 10
- c) Output, 16
- d) Error, sizeof cannot evaluate size of a function

View Answer

Answer: a

Explanation: None.

7. Which of the following cannot be used inside sizeof?

- a) pointers
- b) functions
- c) macro definition
- d) none of the mentioned

View Answer

Answer: d

Explanation: None.

8. What will be the output of the following C code?

- ```
1. #include <stdio.h>
2. (sizeof double = 8, float = 4, void = 1)
3. #define PI 3.14
4. int main()
5. {
6.     printf("%d", sizeof(PI));
7. }
```

- a) Output is 8
- b) Output is 4
- c) Output is 1
- d) Error, we can't use sizeof on macro-definitions

View Answer

Answer: a

Explanation: None.