

Full Marks: 50 Marks

Pass Marks: 20 Marks

1) What is the output of the following code?

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

```
int main() {
```

```
int x, y = 6, z = 6;
```

```
x = y == z;
```

```
printf("%d", x);
```

```
return 0;
```

```
}
```

A) 0

B) 1

C) 6

D) compiler error

2) What is the output of the following program?

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

```
void main() {
```

```
int a=5, b=6;
int result1 = a | b;
int result2 = a & b;
printf("%d and
%d", result2, result1);
}
```

A) 5 and 6

B) 7 and 4

C) 4 and 7

D) compile error

- Bitwise OR (|): It sets each bit of the result to 1 if at least one of the corresponding bits in **a** or **b** is 1. In binary, 5 is **101** and 6 is **110**. Performing bitwise OR results in **111** (7 in decimal).
- Bitwise AND (&): It sets each bit of the result to 1 only if both corresponding bits in **a** and **b** are 1. In binary, performing bitwise AND results in **100** (4 in decimal).

3) What is the output of the following program?

```
#include <stdio.h>
int f(int a, int b){
return a*2 + b%5;
```

```
}  
  
void main() {  
    int a=5, b=7;  
    printf("%d",f(a,b));  
}
```

A) 12

B) 10

C) 5

D) compile error

- $a * 2: 5 * 2 = 10$
- $b \% 5: 7 \% 5 = 2$

Adding these results together: $10 + 2 = 12$.

4) Which of the following declarations is not supported by the C language?

A) String str;

B) char *str;

C) float str = 3e2;

D) Both “String str;” and “float str = 3e2;”

5) 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

6) What is the sizeof(char) in a 32-bit C compiler?

a) 1 bit

b) 2 bits

c) 1 Byte

d) 2 Bytes

7) What is the output of the following code?

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

```
int main(){
```

```
int i=10;
```

```
printf("%d",i++);
```

```
return 0;
```

```
}
```

a) 10

b) 11

c) 12

d) Error

`printf("%d", i++);`: Prints the current value of `i` (which is 10) using `printf` and then increments `i` by 1 due to the post-increment operator.

8) What is the output of the following program?

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

```
int main()
```

```
{  
int y = 10000;  
int y = 34;  
printf("Hello World! %d\n",  
y);  
return 0;  
}
```

- a) Hello World! 34
- b) Hello World! 10000
- c) Compile Time Error**
- d) 34

9) What is the output of the following program?

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

```
void main()
{
int x = 5 * 9 / 3 +
9;
}
```

a) 24

b) 3.75

c) 3

d) 19

10) What is the output of d, a, b?

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

```
//Program 1
```

```
int main()
{
int d, a = 1, b = 2;
d = a++ + ++b;
```

```
printf("%d, %d, %d", d, a,  
b);  
}
```

a) 5,2,3

b) 4,2,3

c) 4,1,3

d) Error

11) Which is the correct format specifier for double type value in C?

a) %LD

b) %ld

c) %lf

d) %LF

13) Increment (++) and decrement (--) are the ____ operators in C?

- a) **Unary**
- b) Binary
- c) Ternary
- d) None of the above

14) Which is the correct format specifier for long long int?

- a) %llf
- b) %ld
- c) %LLD
- d) **%lld**

15) How many bytes does float take?

- a) 8 bytes
- b) 10 bytes

- c) 1 byte
- d) 4 bytes**

16) Which is the correct way for initializing an array of char?

- a) Char array[] ="Sita";
- b) char array[]= 'Sita';
- c) char array[10] = "Sita";**
- d) char array="Sita";

17) What is the output of the program?

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

```
#include <stdbool.h>
```

```
int main() {
```

```
while (true) {
```

```
printf("Ram");
```

```
}
```

```
return 0;
```

```
}
```

a) Ram

b) Compile error

c) Ram is printed unlimited number of times

d) None of the above

18) Processor Directive in C language starts with?

a) \$ symbol (Dollar)

b) @ symbol (at the rate)

c) & symbol (ampersand)

d) # symbol (hash)

19) What is the output of the program?

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

```
int main()
{
    short i;
    for (i = 1; i >= 0; i++)
        printf("%d\n", i);
}
```

- 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**

21) What is the output of the following program?

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

```
float isOdd(int n){  
    if(n%2==1){  
        return 1;  
    }else{  
        return 0;  
    }  
}  
  
void main() {  
    printf("%f",isOdd(7));  
}
```

- a) true
- b) 1
- c) 1.000000**
- d) None

21) What is the purpose of a function prototype in C language?

- a) To declare the function name and its return type.

- b) To define the implementation of a function.
- c) To declare the return type of the function.
- d) To declare the function name, its parameter types and return type.**

22) What is the difference between call by value and call by reference in C language?

a) Call by value passes the address of the argument to the function, while

call by reference passes a copy of the argument value.

b) Call by value passes a copy of the argument value to the function, while

call by reference passes a reference to the argument variable.

c) Call by value and call by reference are the same thing.

d) Call by value passes a reference to the argument variable, while call by

reference passes a copy of the argument value.

23) Which of the following is true about recursive functions in C language?

a) A recursive function cannot call itself.

b) A recursive function must have a base case to terminate the recursion.

c) A recursive function must have a return type of void.

d) A recursive function cannot be called from another function.

24) Which of the following is true about function arguments in C language?

a) Function arguments are always passed by value.

b) Function arguments are always passed by reference.

c) Function arguments can be passed by value or by reference.

d) Function arguments can be passed by value, but not by reference

25) What is the output of the following program?

```
#include <stdio.h>

void funct(int *a){
    *a= *a * 5;
    printf("*a=%d ",*a);
}

int main(){
    int a=6;
    funct(&a);
    printf("a=%d ",a);
}
```

a) a=30 *a=30

b) a=6 *a= 6

c) a= 6 *a=30

d) a=30, *a= 6

- Initially, a is set to 6.

- `funct(&a);` is called, which multiplies the value pointed to by `a` (i.e., 6) by 5. So, `*a` becomes 30, and the function prints `"*a=30 "`.
- Finally, `printf("a=%d ", a);` prints the value of `a`, which remains 30.

26) What is the output of the following program?

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

```
void funct(int a){
```

```
    a= a*6;
```

```
    printf("a in funct is %d,",a);
```

```
}
```

```
int main(){
```

```
    int a=6;
```

```
    funct(a);
```

```
    printf("a in main function is %d ",a);
```

```
    return 0;
```

```
}
```

a) a in funct is 6,a in main function is 6

b) a in funct is 36,a in main function is 36

c) a in funct is 36,a in main function is 10

d) a in funct is 36,a in main function is 6

- Initially, `a` in `main` is set to 6.
- `func(a);` is called, and the value of `a` is passed to the function. Inside the function, `a` is multiplied by 6 (resulting in 36) and printed.
- Finally, `printf("a in main function is %d ", a);` prints the value of `a` in the `main` function, which remains 6.

27) What is the output of the following program?

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

```
int funct(int a){
```

```
    a= a*6;
```

```
    printf("a in funct is %d,",a);
```

```
    return a;
```

```
}
```

```
int main(){
```

```
int a=6;  
a= funct(a);  
printf("a in main function is %d  
",a);  
return 0;  
}
```

a) a in funct is 6,a in main function is 6

b) a in funct is 36,a in main function is 36

c) a in funct is 36,a in main function is 10

d) a in funct is 36,a in main function is 64

- Initially, `a` in `main` is set to 6.
- `funct(a);` is called, and the value of `a` is passed to the function. Inside the function, `a` is multiplied by 6 (resulting in 36), printed, and returned.
- The returned value (36) is assigned to `a` in the `main` function.
- Finally, `printf("a in main function is %d\n", a);` prints the value of `a` in the `main` function, which is now 36.

28) 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

29) The value obtained in the function is given back to main by using _____ keyword.

- a) static
- b) return**
- c) new
- d) volatile

30) What will be the value of x?

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

```
void solve() {
```

```
int x = printf("Shyam");
```

```
printf(" %d", x);
```

```
}
```

```
int main() {
```

```
solve();
```

```
return 0;
```

```
}
```

a) Shyam 5

b) 5

c) Shyam 1

d) 0

- `printf("Shyam");`: Prints the string "Shyam" and returns the number of characters printed, which is 5.
- `int x = printf("Shyam");`: Stores the value (5) returned by `printf` in the variable `x`.
- `printf(" %d", x);`: Prints the value of `x`.

31) What will be the value of a and b?

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

```
void swap(int *a, int *b) {
```

```
int t = *a;
```

```
*a = *b;
```

```
*b = t;
```

```
}
```

```
void function() {
```

```
int a = 3, b = 5;
```

```
swap(&a, &b);
```

```
printf("%d %d", a, b);
```

```
}
```

```
int main() {
```

```
function();
```

```
return 0;
```

```
}
```

a) 3 5

b) 5 3

c) 5 5

d) 3 3

32) What will be the value of a?

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

```
void function() {
```

```
char ch[10] =
```

```
"abcdefghij";
```

```
int a= 0;
```

```
for(int i = 0; i < 10; i++) {
```

```
a += (ch[i] - 'a');
```

```
}
```

```
printf("%d", a);
```

```
}
```

```
int main() {
```

```
function();  
return 0;  
}
```

a) 100

b) 36

c) 45

d) 20

- For the character 'a', the difference is 0.
- For the character 'b', the difference is 1.
- For the character 'c', the difference is 2.
- And so on...

So, the value of **a** will be the sum of the differences, which is $0 + 1 + 2 + \dots + 9$.

The sum of the first 10 natural numbers is given by the formula $(n * (n + 1)) / 2$, where n is the number of terms. In this case, n is 9.

Therefore, the value of **a** will be $(9 * (9 + 1)) / 2 = 45$.

33) What will be the output of the program?


```
#include <stdio.h>

void function() {
    int first = 10, second =
    20;
    int third = first + second;
    {
        int third = second -
        first;
        printf("%d ", third);
    }
    printf("%d", third);
}

int main() {
    Function();
    return 0;
}
```

a) 10 20

b) 10 30

c) 10 20

d) Compilation error

34) What will be the output of the program?

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

```
void function() {
```

```
int a = 3;
```

```
int res = a++ + ++a + a++ + ++a;
```

```
printf("%d", res);
```

```
}
```

```
int main() {
```

```
function();
```

```
return 0;
```

```
}
```

a) 12

b) 20

c) 21

d) 24

1. `a++`: Post-increment, uses the current value of `a` (3) and then increments it by 1. So, `a` becomes 4, and this part contributes 3 to the sum.
2. `++a`: Pre-increment, increments the current value of `a` (4) and then uses it. So, `a` becomes 5, and this part contributes 5 to the sum.
3. `a++`: Post-increment, uses the current value of `a` (5) and then increments it by 1. So, `a` becomes 6, and this part contributes 5 to the sum.
4. `++a`: Pre-increment, increments the current value of `a` (6) and then uses it. So, `a` becomes 7, and this part contributes 7 to the sum.

Now, let's sum up these values: $3 + 5 + 5 + 7 = 20$.

35) What is the correct way to write actual parameters?

- a) `Int sum=sumofnum(a,b);`
- b) `int sum = sumofnum(int a,int b);`
- c) `int sum = sumofnum(a, b);`**
- d) none

36) What is the output of the following program?

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

```
int main() {
```

```
char str1[30]="Harry";  
char str2[30];  
str2 = str1;  
printf("%s",str2);  
}
```

a) Harry

b) None

c) 6fdc00

d) Error occurred

37) What is the output of the following program?

```
#include <stdio.h>  
  
void main() {  
    int a[3] = {5};  
    int i;  
    for(i=0;i<3;i++){  
        printf("%d",a[i]);  
    }  
}
```

}

a) 555

b) 500

c) 5(garbage)(garbage)

d) (garbage)(garbage)5

38) What is the output of the following program?

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

```
int *p;
```

```
int main()
```

```
{
```

```
int i = 5;
```

```
p = &i;
```

```
printf("%d, %x, %d",i,p,*p);
```

```
return 0;
```

```
}
```

a) 5, 62fe1c, 5

b) 5, 433455, 5

c) 5, 5, 433455

d) 5, 5, 5

39) What is the output of the following program?

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

```
int main()
```

```
{
```

```
int a[5] = {11,22,33,44,55};
```

```
int *p;
```

```
p = &a[0];
```

```
p++;
```

```
printf("%d, %d",*p,*(p+1));
```

```
return 0;
```

```
}
```

a) 11, 22

b) 22, 33

c) 11, 33

d) 22, 44

1. `int a[5] = {11, 22, 33, 44, 55};`: Initializes an integer array `a` with five elements: 11, 22, 33, 44, 55.
2. `int *p;`: Declares an integer pointer `p`.
3. `p = &a[0];`: Assigns the address of the first element of the array `a` to the pointer `p`. Now, `p` points to the first element, which is 11.
4. `p++;`: Increments the pointer `p` by 1. Now, `p` points to the second element of the array, which is 22.
5. `printf("%d, %d", *p, *(p + 1));`: Prints the value of the element pointed to by `p` (which is 22) and the value of the next element in the array (which is `*(p + 1)`). In this case, `*(p + 1)` refers to the third element of the array, which is 33.

40) In c programming, `sizeof()` function returns:

a) The length of an array in integer

b) The bytes size required to allocate in memory

c) The array values one by one

d) None of the above

41) An array in C cannot be initialized by which of the following statements?

a) `char a[]= "Hello";`

b) `char a[6]= {};`

c) `char a[6]= {0};`

d) `char a[6];`

`a= "Hello";`

42) What is the output of the following program?

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

```
int main(){
```

```
int A[5]= {1,2,3,4,5};
```

```
printf("%d",sizeof(A)/sizeof(A[0]));
```

```
}
```

a) 6

b) 5

c) 0

d) 1

43) Array is passed into the function using pass by _____.

a) reference

b) value

c) both

d) none of the above

44) What is the output of the following program?

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

```
int main(){
```

```
int A[5]= {1,2,3,4,5};
```

```
int *p;
```

```
p= A;  
printf("%d",*(p+1));  
return 0;  
}
```

- a) 1
- b) 3
- c) 2**
- d) 5

45) How to find the length of an array in C?

- a) sizeof(a[0])
- b) sizeof(a)/sizeof(a[0])**
- c) sizeof(a)*sizeof(a[0])
- d) sizeof(a)

46) How to initialize a list of strings in an array?

- a) Char [5][20]={“ram”,“shyam”,“hari”,“sita”,“gauri”};

b) `char array[20][5]={“ram”,“shyam”,“hari”,“sita”,“gauri”};`

c) `char`

`array[5][20]={“ram”,“shyam”,“hari”,“sita”,“gauri”};`

d) `char array[5][20]={ram,shyam,hari,sita,gauri};`

47) What will be the output of the program?

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

```
void main(){
```

```
int *ptr=10;
```

```
printf("Value of ptr:%d",*ptr);
```

```
}
```

a) 10

b) Segmentation fault

c) Compilation error

d) 0

48) What will be the output of the program?

```
#include <stdio.h>

#include <string.h>

void main(){

char name1[10]="ram";

char name2[10]="Shyam";

printf("String compare
result:%d",strcmp(name1,name2));

}
```

a) -1

b) 1 or 31

c) 0

d) -33

- `char name1[10] = "ram";`: Initializes a character array `name1` with the string "ram."
- `char name2[10] = "Shyam";`: Initializes a character array `name2` with the string "Shyam."
- `printf("String compare result: %d", strcmp(name1, name2));`: Compares the strings using `strcmp` and prints the result.

The `strcmp` function returns an integer:

- If the result is 0, it means the strings are equal.
- If the result is negative, it means the first string is lexicographically less than the second.

- If the result is positive, it means the first string is lexicographically greater than the second.

In this case, the result depends on the lexicographical order of "ram" and "Shyam." The output will be non-zero, indicating that the strings are not equal.

49) What will be the output of the program?

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

```
#include <string.h>
```

```
void main(){
```

```
char name1[10]="ram";
```

```
char name2;
```

```
printf("String copy result:%s",strcpy(name1,name2));
```

```
}
```

a) Segmentation fault

b) ram

c) Compilation error

d) Runtime error

50) In C, what is the value of a pointer variable if it hasn't been assigned any address?

a) 0

b) Null

c) undefined

d) 1