Q.1) Subject: C Programming Max Marks: 1

In the given C program choose the correct option for the above code with the correct reason.

```
#include<stdio.h>
int main()
{
        char *p="hello";
        char *q="hello" ;
        if (p==q)
        printf("hello");
        else
        printf ("not hello");
        return 0;
}
```

A **hello, because Returning address of same string.** Correct Option

Solution: (A)

Solution: Ans is (A). hello, because Return address of same string.

Here, we have declared two character pointers p and q and initializing these two pointers with the same string.

char *p="hello"; // p is the character pointer which is pointing to memory location where string " hello" is stored.

char *q="hello"; // q is the character pointer, which will start pointing to memory location where string " hello" is stored. So it will take the same address value.

if (p==q) // here, we are comparing two addresses.

hello is already written by p when again the same string comes the compiler returns the address of available string. So output of program is hello. If we again write **"hello"** as next line, then it first check that if given String constant is present in String pooled area or not. If it presents then it will point to it, otherwise creates a new String constant.

B **not hello because Returning different addresses.**

C **compile time error**

D **run time error**

Q.2) Subject: C Programming Max Marks: 1

For declaration of different dimensional arrays, Choose the option which is not correct.

A **int a[3][4][5] ;**

B **int a[10];**

C **int a[ ] [4][5];** Correct Option

Solution: (C)

Solution: Answer is int a[ ] [4] [5];

An array is a collection of similar type of data items and each data item is called an element of the array. When we declare different types of arrays then we need to give their sizes too. It means we can not leave subscripts blank. With this condition, check with options:

int a[3][4][5] ; // this is a three dimensional array in which there are 3 - two dimensional arrays of 4 rows and 5 columns

int a[10]; // this is single dimensional array with 10 elements.

int a[ ] [4][5]; // There are some two dimensional arrays of 4 rows and 5 columns. Here, number of 3D arrays are unknown so this is incorrect way of array declaration.

So, answer is int a[ ] [4] [5];

D **None of these**

Q.3) Subject: C Programming Max Marks: 1

What will be the output of the following given C program.

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

```
int main()
{
    int a=210,b=120,c;
    c=a>b?1,2,3:2,5,6,7;
    printf("%d",c);
}
```

Choose the correct option..?

A 1

B 2

C 3 — Correct Option

Solution: (c)

Solution: Answer is 3.

Here, first condition will be checked and if condition is true then value before colon is chosen and if-condition is false then it will print value after colon.

int a=210,b=120,c; // here a, b, c are declared as integer and a and b also assigned with values as a=210, b=120

c=a>b?1,2,3:2,5,6,7; // C = a>b = 210>120 . Condition is true. Then expression before colon will be evaluated.

The compiler takes the last value. That's the comma operator, and it evaluates its operands left-to-right and returns the rightmost one. So the last value in case of true condition is 3. So 3 will be picked and assigned to c.

printf("%d",c);// c value is printed 3.

So answer is 3.

D 7

Q.4) Subject: C Programming Max Marks: 1

What will be the output of the following program.

```
#include<stdio.h>
int main()
{
    double *ptr = (double*)100;
    ptr = ptr+2;
    printf("%ld", ptr);
    return 0;
}
```

A 100

B 112

C 114

D 116 — Correct Option

Solution: (D)

Solution: Answer is 116

double *ptr = (double*)100; // here pointer ptr as double with value 100 is declared and value is also typecasted with double ptr = ptr+2; // Double type has size 8 Byte so double pointer step size is 8. If we are adding 2 in step pointer, it will add 16 to the address of double as

100 + 16 =116

printf("%ld", ptr); // it will print 116.

So, answer is 116.

Q.5) Subject: C Programming Max Marks: 1

choose the correct option for the given following given code.(Note: size of pointer variable in 32-bit machine is 4 and in 64 bit machine is 8.) And here we are compiling on 64-bit machine.

```
#include<string.h>

int main()

{

      char *p="hello";

      char q [] ="hello" ;

      printf("%d %d %d", sizeof(*p), sizeof(p), sizeof(q));

      printf("\t %d %d", strlen(p), strlen(q));

      return 0;

}
```

A 1 4 5 4 4

B 1 4 6 5 5

C 1 8 5 5 5

D 1 8 6 5 5 Correct Option

Solution: (D)

Solution: Ans is (D) 1 8 6 5 5

Sizeof operator returns no. of bytes consumed by its operand. And size of pointer variable in 64 bit machine is 8.

sizeof(*p): returns the size of type what the pointer points to. So here it is pointing to character pointer so it will be 1.

sizeof(p): printing size of it as it is pointer type. So size of p will be 8 in 64-bit machine.

sizeof(q): it is printing size of q and q is the character array. And it will store as

h	e	l	l	o	\0

So its size will be 6.

strlen(p), strlen(q): printing length of the string as string is containing 5 characters so answer will be 5.

Q.6) Subject: C Programming Max Marks: 1

What is the output of the given code.

```
#include<stdio.h>
int main()
{
            int b=65;
            void p=b;
            int c= (int)p;
            printf("%d\n",p);
            return 0;
```

A 65

B A

C Produce error Correct Option

Solution: (C)

Solution: Ans is (C). Produce error

```
int b=65; //declare b as integer with value 65
```

`void p=b; // declaring p as void and assigning b to it.`

This line will produce error as void data type is not allowed because size of void data type is not known. So compile time error is arising in this program because of this line. Therefore option (c) is correct.

D **Prints nothing**

Q.7) Subject: C Programming Max Marks: 1

What will be the output of the following given C program.

```
#include<stdio.h>

int main(void)

{

    int a;

    a = (1, 2, 3);

    printf("%d", a);

    return 0;

}
```

A **1**

B **2**

C **3** Correct Option

Solution: (c)

Solution: Answer is 3.

In this program, brackets are used so comma operator is executed first and we get the output as 3. As assignment operator works from right to left.

int a; // we are declaring a as integer

a = (1, 2, 3); // Here, we are using brackets so expression at the right side is treated as a whole means one expression. As we are using comma operator, so this expression will pick right most element and that value will be assigned to a.

printf("%d", a); // That's why 3 will be printed here.

Note: if we don't use brackets here, then it will simply take 1 because at that time all three will be treated as different expressions.

D **Compiler error**

Q.8) Subject: C Programming Max Marks: 1

Consider the following three statements of C program.
statement 1: int a;
statement 2: extern int b;
statement 3: static int c;
which of the following option is correct.

A **statement 1, statement 2 and statement 3 only declare variables, don't define them**

B **statement 1 and statement 2 declare and define variables while statement 3 only declares, don't define.**

C **statement 1 and statement 3 declare and define variables while statement 2 only declares, don't define.** Correct Option

Solution: (c)
Solution: Ans is statement 1 and statement 3 declare and define variables while statement 2 only declares, don't define.

Declaration of a variable is for informing to the compiler the following information: name of the variable, type of value it holds and the initial value if any it takes. i.e., declaration gives details about the properties of a variable. Whereas, **Definition** of a variable says where the variable gets stored. i.e., memory for the variable is allocated during the definition of the variable. But in C language definition and declaration for a variable takes place at the same time. i.e. there is no difference between declaration and definition.
if we want to only declare variables and not to define it i.e. we do not want to allocate memory, then we can do this with the help of extern keyword as it is done in statement 2. *Static variables* (like global *variables*) are initialized as 0 if not initialized explicitly. when a variable is declared it contains undefined value that is known as garbage value. So
statement 1: int a; // declare and defined with garbage value
statement 2: extern int b; // only delcared
statement 3: static int c; // declared and initialized with 0

D **all the statements declare and define the variables.**

Q.9) Subject: C Programming Max Marks: 1

In the given following program, what is returned by the final printf. If input is given as 10 20.

```
#include<stdio.h>

int main()

{

          int p, r;

          int q =scanf("%d %d", &p, &r);

          printf ("%d", q+ printf ("AAIC"));

return 0;

}
```

A returning q+ AAIC.

B Returning q AAIC

C returning all the numbers printed by it on the display.

D Returning number of characters printed by it on the display. Correct Option

Solution: (D)

Solution: Ans is (D) returning the number of characters printed by it on the display.

int p, r; // declaring p and r as integer type

int q =scanf("%d %d", &p, &r);// declaring q as integer type and assigning it scanf() function

printf ("%d", q+ printf ("AAIC")); // printf ("AAIC") = AAIC

printf ("%d", q+AAIC); // it will count 2 input values and 4 characters of string so output will be printed AAIC6.

So we can say that it is returning the number of characters printed by it on the display.

Q.10) Subject: C Programming Max Marks: 1

What will be the output of the following c program.

```
#include<stdio.h>

int main()

{

    int p=1;

    switch (p)

              {

                        case 1: printf("hi");

                        case 5-4: printf("bye");

              }

              return 0;

   }
```

A hi

B hi bye

C duplicate case value Correct Option

Solution: (C)

Solution: Ans is (C) duplicate case value.

`int p=1; // p is declared and initialized with value 1.`

`switch (p)// switch(1) will be executed and it will search for a case which has the same value as this switch expression has.`

`case 1: printf("hi"); // it is case 1.`

`case 5-4: printf("bye"); // it is also case 1.`

Here, cases are given as case1 in which case is denoted by 1 and second case 5-4. So here after solving expression, this case will also be case1 so

here two cases are with the same integer value so error: duplicate case value is produced.

D **run time error**

Q.11) Subject: C Programming Max Marks: 2

What will be true statement for the given code.

```
#include<stdio.h>
int main()
{
        int i=5;
        int j;
        i=j=4;
        printf("%d", i);
        if (j==(i=0))
            printf ("APPLIED");
        else
            printf ("GATE");
}
```

A **Program will be executed successfully with left to right associativity.**

B **There will be compile time error due to associativity confusion.**

C **There will be run time error in the program**

D **Program will be executed successfully with right to left associativity.** Correct Option

Solution: (D)

Solution: Answer is Program will be executed successfully with right to left associativity.

Here, assignment operator works with right to left associativity. It returns right hand side evolution.

int i=5; // i is declared as integer type and assigned with 5.

int j; // j is declared.

i=j=4; // as we know that assignment operator works from right to left so it will be executed as:

Here, j value will be assigned as j=4.

Then i= 4

printf("%d", i); // Here i value 4 is printed

if (j==(i=0)) // Here, j=4 and i value will become 0 and then it is compared with j then condition is not true

printf ("APPLIED"); // skipped as condition is false.

else

printf ("GATE"); // this will be printed.

Therefore, output of the program will be 4GATE.

Q.12) Subject: C Programming Max Marks: 2

What will be the output of the following C code.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[]= "APPLIED";
    const char *ptr;
    ptr=str;
    ptr++;
    printf("%s", --ptr);
    return 0;
}
```

A **APPLIED** Correct Option

Solution: (A)

Solution: Ans is (A) APPLIED

Here we can be confused by seeing const and pointer together but we are fetching value from address so it is possible to access. Let character array is starting from 1000.

1000	1001	1002	1003	1004	1005	1006
A	P	P	L	I	E	D

char str[]= "APPLIED";// character array is given in str

const char *ptr; // ptr is constant character pointer

ptr=str; // here str is assigned to ptr so ptr = 1000

Ptr++; // here we are moving ptr forward by 1 in character array str i.e. 1001

printf("%s", --ptr);// here, we are moving back by 1 in character array i.e. 1000

So it will print APPLIED.

B **PPLIED**

C **PLIED**

D **LIED**

Q.13) Subject: C Programming Max Marks: 2

Which C program will be executed successfully.

(i)
```
#include<stdio.h>

   int main(void)
{
    int a = 1, 2, 3;
    printf("%d", a);
    return 0;
}
```

(ii)
```
#include<stdio.h>

int main(void)

{

  int a;

     a = 1, 2, 3;

printf("%d", a);

return 0;

}
```

A **Only (i)**

B **Only (ii)** Correct Option

Solution: (B)

Solution: Answer is Only (ii)

We will review (i) and (ii) program line-by-line. and then we try to get where and which code line is making code errorneous.

```
#include<stdio.h>
int main(void)
{
  int a = 1, 2, 3;
  printf("%d", a);
  return 0;
}
```

This program will give error as Error:expected unqualified -id before numeric constant. Because here, we are trying to assign multiple numeric values to a single variable that is incorrect. Initialization and declaration can not be done together with multiple values. That's why, code a is giving error at compile time.

```
#include<stdio.h>

int main(void)

{

int a; // declares variable a as integer type.

a = 1, 2, 3; // Here 1 is assigned to a, as all these three  are treated individually.
```

printf("%d", a); // 1 will be printed here.

return 0;

}

So, this code will be executed successfully and will print 1 as output.

Answer is Only (ii) will be executed successfully.

C **(i) and (ii)**

D **None of the above**

Q.14) Subject: C Programming Max Marks: 2

What will be the output of this code, if input is 3 and 3.

```
#include<stdio.h>

int main()

{

          int p, r;

          int q =scanf("%d %d", &p, &r);

          printf ("%d", q+ printf ("AAIC"));

return 0;

}
```

A **33+AAIC**

B **33 AAIC**

C **AAIC6** Correct Option

Solution: (c)

Solution: Ans is (C) AAIC6

int p, r; // declaring p and r as integer type

int q =scanf("%d %d", &p, &r);// declaring q as integer type and assigning it scanf() function. p and r will be given with values p=3 and r=3

printf ("%d", q+ printf ("AAIC")); // printf ("AAIC") = AAIC

printf ("%d", q+AAIC); // it will count 2 input values and 4 characters of string so output will be printed AAIC6.

As it is considering two integer inputs as two and then AAIC string is printed then total 6 characters is being displayed on the screen so the final output will be AAIC6.

D **6AAIC**

Q.15) Subject: C Programming Max Marks: 2

What will be the output of the following code.

```
#include<stdio.h>

int main()

{

int p = printf("");

switch (p)

{

   case 0: printf ("GATE");

   default: printf (" APPLIEDCOURSE");

 }

return 0;

}
```

A **Compile time error**

B **Run time error**

C **APPLIEDCOURSE**

D **GATE APPLIEDCOURSE** Correct Option

Solution: (D)

Solution: Answer is GATE APPLIEDCOURSE

The printf() function is used for printing the output. It returns the number of characters that are printed. If there is some error then it returns a negative value.

int p = printf("");// Here, 0 length string is given so it prints 0

switch (p)// here switch(0)will be executed

case 0: printf ("GATE"); // So it will go to case 0.

default: printf (" APPLIEDCOURSE"); // and then will go to default case also as there is no break statement between these two cases.

So, it will print GATE APPLIEDCOURSE.

close