OVERALL ANALYSIS Solution Report

All Correct Answers Wrong Answers Not Attempted Questions

Q.1) Max Marks: 1

Choose the correct option from the following statement.

A global static variables takes some garbage values if not initialized.

B Initialization of static variables happen only when first time they called. Correct Option

Solution: (B)

Solution: Ans is (B). Initialization of static variables happen only when first time they called. And second time if we call, they are already presented in the memory. They are not initialized and no memory is created.

C Initialization of static variables happen every time they called.

D Lifetime of static variables are from the time function called and till the time function ends.

Q.2) Max Marks: 1

Which char will be output of the following given code. When 'A' ascii value is 65

```
#include<stdio.h>

int main()

{

      char a[] = { 'G', 'A', 'T', 'E', '\n', 'P','\0'};

      char *b, *c;

      b= a+3;

      c = a;

      printf("%c", ++ *b+*c++-63);

      return 0;

}
```

A T

B E

C N Correct Option

Solution: (C)

Solution: Ans is (C). N

= ++ *b+*c++-63 // here *b and *c means content of b and c that is E and G

= ++ (E) + (G++) -63

= ++ (69) + (71++) -63 // we are writing ascii value of E and G

= 70+71-63

= 78 which is N

D S

Q.3) Max Marks: 1

what will be the output of the following code when we give input as "appliedcourse".

```
#include "stdio.h"

int main()

{

    char arr[100];

    printf("%d\t", scanf("%s", arr));
```

```
    printf("%d", printf("%d",printf("GATE")));
    return 0;
}
```

A **1 GATE41** — Correct Option

Solution: (A)
Solution: Ans is (A)1 GATE41
In C, printf returns number of input taken by the scanf function inside it. So it will simply return 1 not number of characters. So output will be 1
And in second printf, we are using printf inside printf so inside printf will print GATE and then second inside printf will print the number of characters in previously returned output and then outer printf will also print number of characters printed by last printf that was 1 only so it will print 1 so output will be 1 GATE41.

B **13 GATE4**

C **13 GATE41**

D **1 GATE4**

Q.4) Max Marks: 1

What will be the correct option for the following given program.

```
#include <stdio.h>

int *fun()

{

        int b=10;

        b++ ;

        return 0;

}

int main()

{

        int *p;

        p=fun();

        printf("%d\n", p);

        return 0;

}
```

A **10**

B **0** — Correct Option

Solution: (B)

Solution: Answer is 0.
Here, we are simply confusing by giving pointer function and manipulating that value but at last we are returning 0 so it will simply give 0 in output. Here, we will start from the main()

`int main()`
`int *p; //here, we have declared`
`p=fun();// we are assigning function pointer to the integer pointer and calling fun()`	

	`int *fun()// pointer function`
	`int b=10;// declare b as integer and assigned value as 10.`
	`b++ ; // post-increment b.`
	`return 0;// returning 0.`

Coming back to main() printf("%d\n", p);// print 0	

So, here correct answer is 0.

C Address of b

D Error

Q.5) Max Marks: 1

Choose the incorrect statement.

A Any variable which is declared inside curly braces is auto by default.

B Any variable which is auto, initialized to zero if we don't explicitly initialize it. Correct Option

Solution: (B)
Solution: Ans is (B). Any variable which is auto, initialized to zero if we don't explicitly initialize it.
This is incorrect as any variable which is auto by default if not initialized with any value then it takes some garbage value.

C Extern keyword is helping to increase the scope or region of code where i can access this variable.

D Any variable which is extern is zero by default.

Q.6) Max Marks: 1

Following C program declaration is done below.

```
union {
        float a;
        struct {
                float b;
                int c;
                char d[9];
        } t;
        union {
                long f;
                int x[4];
        } u2;
}u1;
```

Here different data types are stored in the memory in which char, int, float and long is taking memory as 1 byte, 2bytes, 4bytes and 8 bytes. Then the total size of memory which we require for the above code.

A 4

B 14

C 15 Correct Option

Solution: (C)
Solution: Ans is 15
Union takes memory of size of maximum variable size only and structure takes memory of size for all variables so in the above code first we can see that inside union, there is one float variable, one structure and one union so
float size =4 bytes
struct s size = 4 + 2 + 9 = 15 bytes
union u2 size = 8 bytes
So union u1 size = maximum of these three then =15 bytes

D 16

Q.7) Max Marks: 1

Choose the correct option for the given following code.

```
#include<stdio.h>
int main()
{
  int a,b,c;
    a=b=10;
    b=c=50;
    ("%d %d %d"  a b c);
```

```
    return 0;
}
```

A. It will print 10 50 50

B. It will give error

C. Garbage values will be printed.

D. No value will be printed. — Correct Option

Solution: (D)
Solution: Ans is (D) No value will be printed.
As we haven't written statement with printf so it will not print anything but also it will not produce any error as we have written everything correctly.

Q.8) Max Marks: 1

What will be the output of the following code.

```
int fog(int *a, int *b)
{
      a=b;
      *a= 3;
}
int m=2,n=4;
int main()
{
      fog(&m,&n);
      printf("%d %d", m,n);
      return 0;
}
```

A. 2, 4

B. 4, 2

C. 2, 3 — Correct Option

Solution: (C)

Solution: Ans is (C). 2, 3

As we are starting from main and m ,n are already initialized with values so

let m =2 (let 1000) and n =4 (let 2000)

fog(1000, 2000) → fog(*a, *b) => a=b now a will start pointing to address of b

*a = 3 // changing content of a value as 3 so will make change at address 2000.

So here Ans will be 2,3

D. 3, 4

Q.9) Max Marks: 1

What will be the output of the following code.

```
#include<stdio.h>
int main()
{
      char q[] = "gate", *ptr;
      ptr = q;
      while(*ptr!='\0')
      {
            ++ *ptr++ ;
      }
      printf("%s %s",ptr,q);
      return 0 ;
```

```
}
```

Options are

A gate

B hate

C fauf

D hbuf — Correct Option

Solution: (D)

Solution: Ans is (D). hbuf

```
++ *ptr++  // Here, post increment is used for moving pointer
from one char to next and pre-increment is user for increasing
character value. It is executing as (++(*p))++

And at last ptr is reached to null so it will print nothing. So
output will be   hbuf
```

Q.10) Max Marks: 1

Choose the correct option for the following code.

```
#include <stdio.h>

int f1(int a);

int f2(int b);

main()

{

    int a = 1, b = 2;

    for (int i=1; i<= 2; ++i)

    {

        b *= f1(a) + f2(a);

        printf ("%d ", b);

    }
}

f1(int a)

{

    int b = f2(a);

    return (b);

}

f2(int a)

{

    static int b = 10;

    b += 1;

    return (b+a);

}
```

A Output of program will be 50, 50.

B Output of program will be 145, 145.

C both the values have 14 as their last two digits in output.

D both the values have 50 as their last two digits in output. — Correct Option

Solution: (D)

Solution: Ans is (D). both the values have 50 as their last two digits in output

Here, output of program will be 50 1450

Q.11) Max Marks: 2

For the given C program if we give input for x and y from the keyboard. Then

```
#include<stdio.h>
#define  swap1(a,b)a=a+b;b=a-b;a=a-b;
int swap2(int x,int y);
int main(){
    int x,y ;
    scanf("%d %d", &x,&y);
    swap1(x,y);
    printf("v1=%d  v2=%d  ",x,y);
    printf("v3=%d",swap2(x,y));
    return 0;
}
int swap2(int a,int b)
{
    int temp;
    temp=a;
    a=b;
    b=temp;
    return (a,b);
}
```

For which value of x and y, E=100 is true where E = v1 + 2* v2 + v3. Choose the correct option.

A **x=30 and y=20** — Correct Option

Solution: (A)

Solution: Ans is (A) x=30 and y=20

When we give input as 30, 20. Then x=30 y=20 then swap1(30,20) is replaced with the macro line as

```
a=a+b; a= 30+20, a=50
b=a-b; b=50-20, b=30
a=a-b; a= 50-30, a= 20
so when we print v1= 20 v2=30 will be printed. Then next printf will
      call swap2(20,30)
int temp;
    temp=20;
    a=30;
    b=20;
    return (a,b); // here it will return rightmost value that is b
     and its value is 20 so v3= 20
```

E = v1 + 2* v2 + v3 = 20 + 2*30 + 20 = 100

B **x=20 and y=10**

C **x=10 and y=30**

D **x=30 and y=40**

Q.12) Max Marks: 2

Consider the function func shown below:

```
int func(int num)

{

    int count = 0;

    while (num)

    {

        count++;

        num >>= 1;

    }

    return (count);

}
```

What is returned by func(100)is

A **6**

B **7** — Correct Option

Solution: (B)

Solution: Ans is (B) 7

When we convert 100 to binary then by this function we are printing position of its MSB bit. Here, 100 = 1100100 so MSB at 7th position so its answer will be 7.

C **8**

D **9**

Q.13) Max Marks: 2

what will be the output of the following code.

```
#include<stdio.h>

fun(int *a, int n)

{

      if(n<=0) return 0;

      if(n<=1) return 1;

      else if(*a%3 == 0) return *a- fun(a+1,n-1);

      else return *a+ fun(a+1,n+1);


}

int main()

{

      int a[]= {20,9,12,6, 8};

      printf("%d", fun(a,2));

}
```

A **18** Correct Option

Solution: (A)

```
Solution: Ans is (A). 18
```

Here, in fun(a,2) by using a we are calling array

fun(20,2)

*a = 20 and n=2

check all conditions and come to last else

*a+fun(a+1, n+1)

20 + fun(9,3) // here we moved to the next array element = 20+(-2) = 18

move to the else if 9 -fun (12, 2) =9 -11 = -2

move to else if 12 - fun(6, 1) = 12 -1 =11

when we get n=1 then return 1 back and fill values to previous function calls.

B **12**

C **20**

D **24**

Q.14) Max Marks: 2

What will be the output of the following code

```
#include<stdio.h>
void fun(int *p)
{
      printf("%d \t", *p);
      if (*p > 0)
      {
            *p = *p -2 ;
            fun (p);
      }
      else
      *p = 0;
}

int main()
{
      int i=10, j=9;

      fun(&i);
      printf("%d\n", i);
      fun(&j);
      printf("%d\n", j);
}
```

A Correct Option

```
10864200
97531-10
```

Solution: (A)

Solution: Ans is (A). 10 8 6 4 2 0 0
9 7 5 3 1 -1 0

Fun(10) – 10 – 10>0 – fun (8) – 8 – 8> 0 – fun(6) – 6 – 6>0 – fun(4) – 4 – 4>0
Fun(2) – 2 – 2>0 – fun(0) – 0 – 0> 0

Fun(9) – 9 – 9>0 – fun (7) – 7 – 7> 0 – fun(5) – 5 – 5>0 – fun(3) – 3 – 3>0
Fun(1) – 1 – 1>0 – fun(-1) – -1 – -1> 0 - 0

B

```
97531-10
10864200
```

C

```
10864220
97531-1-1
```

D

```
10864000
9753100
```

Q.15) Max Marks: 2

What will be the sum of the call values (for each function call) at the end of the program.

```
int callme(int x)

{

    static int call =1;

    call++;

    call = call + x;

    printf("%d\n",call);

    return call;

}

int main()

{    int y;

    for(int x=1; x<=3; ++x)

          y = callme(x);

}
```

A 6

B 10

C 19 Correct Option

Solution: (C)

Solution: Ans is (C) 19

when x=1 and 1< =3	when x=2 and 2< =3	when x=3 and 3< =3
y =callme(1)	y =callme(2)	y =callme(3)
go to the body of that function	go to the body of that function	go to the body of that function
static call =1	static call =3	static call =6
call++ = 1++	call++ = 3++	call++ = 6++
call = call+x =2+1=3	call = call+x =4+2=6	call = call+x =7+3=10
return back to the main()	return back to the main()	return back to the main()
++x =2	++x =3	++x =4
		4 <= 3 (condition fails)

then sum of all call values= 3+6+10 = 19

D 24

Q.16) Max Marks: 2

what will be the output of the following code.

```
#include <stdio.h>

#define AAIC(a,i) a+i

#define GATE(a) a*10

int main(){

   int a=3;

    int i=30;

    printf("%d ",AAIC(GATE('a'),i));

    return 0;

}
```

A 60

B 397

C 1000 Correct Option

Solution: (C)

Solution: Ans is (C)1000

Here, we are applying macro and precedence of operators. And we are confusing by putting single quotes on a,

printf("%d ",AAIC(GATE('a'),i)); //prints 1000

GATE('a') = go to the macro **`GATE(a) a*10 =`** 97*10

AAIC(GATE('a'),i) = AAIC(97*10,30) =go to the macro **`AAIC(a,i) a+i`**

= 97*10+30 = 1000

So answer will be (C). 1000

D 1940

Q.17) Max Marks: 2

What will be the output of the following code.

```
int main()

{

    int a=10;

    int static b=20;

    int c=30;

    printf("%d  ", (a+1),(b=a+2), (c=b+3));

     printf("%d",c);

    return 0;

}
```

A 23, 23

B 11, 23 Correct Option

Solution: (B)

Solution: Ans is (B). 11, 23

As per C11 standard, all the arguments of printf are evaluated whether these are printed or not. And these expressions are evaluated individually no value of one expression will affect the value of other expression. And evaluation in printf is done from right to left.

printf("%d ", (a+1),(b=a+2), (c=b+3)); // output of first variable will be printed = 10+1=11

printf("%d",c); // here c value will be printed after expression evaluation that is c=b+3 = 20+3 = 23

C 11, 15

D 22, 23

Q.18) Max Marks: 2

Consider the following c program

```
int a, b, c = 10;

void Fun (void);

int main ()

{

  static int a = 4;

  Fun();

  a += 1;

  Fun();

}

void Fun (void)

{

  static int a = 8;

  int b = 1;

  a += ++b;

  printf (" \t %d %d " , a, b);

}
```

What will be the output of the above code

A 10 2 12 2 Correct Option

Solution: (A)

Solution: Ans is (A) 10 2 12 2

initially a,b, c is declared and also c is initialized.

then in main a= 4 i.e. static

calling fun() in fun also a =8 i.e. static and b=1 a+=++b => a = a+ ++b => a=8+2 =10 print a b as 10 2 then come back to main a+=1 => a=a+1 =4+1=5	again calling fun() in fun a is initialized as static so a = 10 that will remain same as updated in previous call a+=++b => a = a+ ++b but b value will be initialized each time so it will be 1 => a=10+2 =12 print a b as 12 2

So final output will be 10 2

B 10 1 12 1

C 10 2 10 2

D 12 1 12 2

Q.19) Max Marks: 2

what will be the output of the following code.

```
int main()

{

    int a=1;

    int b=1;

    int c= 1;

    int d= a++ && b-- ||--c;

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

    return 0;
```

}

A 1, 1, 1, 1

B 1, 0, 0, 1

C 2, 1, 0,0

D 2, 0, 1, 1 — Correct Option

Solution: (D)

Solution: Ans is (D) 2, 0, 1, 1

Here, logical AND and logical OR is applied with post increment, pre decrement and post decrement.

```
c= a++ && b-- ||--c;

at the time of logical AND, if left operand/expression is 0 then
we simply say its output is 0 without even evaluating right
operand/expression.

But here left expression is non-zero so we will go further after
evaluating it

a++ = 1++ =2

b-- = 1-- = that is post decrement so there will be no change at
the time of execution of current statement.

 so 2 && 1 = 1

At the time of logical OR if left operand/expression is 1 then we
simply say its output is 1 without even evaluating right
operand/expression.

So 1 || --c will directly say print it as 1. So d value willbe 1.
```

Q.20) Max Marks: 2

What will be the output of the following code.

```
#include <stdio.h>

int main()

{

    int x[ ]= { 13, 26, 39, 52, 65} ;

    int *y[ ]= { x+5, x+4, x+3, x+2, x+1};

    int **z = y;

    *z++;

    printf("%d %d %c", z-y, *z-x, **z);

    return 0;

}
```

A garbage, garbage, garbage

B address, address, 13

C 1, 4, 65

D 1, 4, A — Correct Option

Solution: (D)

Solution: Ans is (D). 1,4, A

x values	13	26	39	52	65
Address(let)	1000	1004	1008	1012	1016
	x	x+1	x+2	x+3	x+4

values		1016	1012	1008	1004	1000
y ==	x+5	x+4	x+3	x+2	x+1	x

address(le t)	2000	2008	2016	2024	2032	Not held by array y

int **z = y;// z is double pointer which is holding address of y.

*z++; // move the pointer one position. initially it was at 2000 and now it will point at 2008.

printf("%d %d %c", z-y, *z-x, **z);

z-y = 2008-2000 = 8/8 = 1

*z-x = 1016 - 1000 = 16/4 = 4

**z => *z = 2008 => *(2008) = 1016 = 65 and printing it as %c then ascii value of 65 = A

so output will be 1, 4, A

close