Q.21 void main ( )

```
{
    int a[2][3][2] = {1,2,3,4,5,6,7,8,9,10,11,12};
    printf("%u %u %u %u", a, *a, **a, ***a);
    printf ("%u %u %u %u", a+1, *a+1, **a+1, ***a+1);
}
```

Assuming the base address of the array to be 1000 and the size of integer is 4 bytes, what are the values printed by printf statements

Q.22 What is printed by the following C program?

```
#include <stdio.h>
int f(int x, int *py, int **ppz)
{
    int y, z;
    **ppz += 1;
    z  = **ppz;
    *py += 2;
    y = *py;
    x += 3;
    return x + y + z;
}
void main()
{
    int c, *b, **a;
    c = 4;
    b = &c;
    a = &b;
    printf( "%d", f(c,b,a));
    getchar();
}
```

Q.23 What does the following C-statement declare?

int (* f) (int *);

(A) A function that takes an integer pointer as argument and returns an integer

(B) A function that takes an integer as argument and returns an integer pointer

(C) A pointer to a function that takes an integer pointer as argument and returns an integer.

(D) A function that takes an integer pointer as argument and returns a function pointer

Q.24 Consider the following C program

```
# include < stdio.h>
int main ( )  {
    static int a[ ] = {10, 20, 30, 40, 50};
```

```
        static int *p[ ] = {a, a+3, a+4, a+1, a+2};
    int **ptr = p;
    ptr++;
    printf ("%d%d", prt–p, **ptr);
}
```

The output of the program is_____.

Q.25 Consider the following snippet of a C program. Assume that swap (&x, &y) exchanges the contents of x and y.

```
int main ( )
  {
int array [ ] = {3, 5, ,1, 4, 6, 2};
int done = 0;
int i;
while (done = = 0)
   {
done = 1;
for (i = 0; < = 4; i ++)
  {
if (array[i] < array[i+1])
  {
  swap (& array [i], & array [i+1]);
done = 0;
  }
  }
for (i = 5; i > = 1; i – –)
  {
if (array[i] > array [i –1])  {
swap(&array[i], &array[i–1]);
done = 0;
  }
  }
  }
printf("%d", array [3]);
  }
```

The output of the program is _____.

Q.26 Consider this C code to swap two to integers and these five statements: the code

```
        void  swap (int * px, int * py )
{
        * px = *px –*py;
        * py= *px +* py;
        *px = *py –px;
}
```

S1 : will generate a completion error

S2 : may generate a segmentations fault at runtime depending on the arguments passed

S3 : correctly implements the swap procedure for all input pointers refereeing to integers stored in memory locations accessible to the process

S4 : implements the swap procedure correctly for some but not all valid input pointers

S5 : may add or subtract integers and pointers

(A) S1 (B) S2 and S3 (C) S2 and S4 (D) S2 and S5

Q.27 Consider the following C program.

```
# include <stdio. h>
int main ( )  {
int a [4] [5] = { {1,2,3,4,5},
                  {6,7,8,9,10},
                  {11,12,13,14,15},
                  {16,17,18,19,20} };
print f  ("%d \ n",*(*a + ** a + 2) + 3) );
return (0);  }
```

The output of the program is _____.

Q.28 Consider the following program :

```
int f (int *p, int n)
   {
    if (n < = 1) return 0;
    else
                  return max (f (p +1, n-1), p[0]-p [1]);
  }
int main ( )  {
    int a [ ] = {3, 5, 2, 6, 4};
    printf ("%d", f(a, 5));
    }
```

Note : max (x, y) returns the maximum of x and y. The value printed by this program is _____.

Q.29 Consider the following C program segment.

```
#include <stdio.h>
int main ( )
{
char s1 [7] = "1234", *p;
p = s1 + 2;
*p = '0';
printf ("% s", s1);
}
```

What will be printed by the program?

(A) 12 (B) 120400 (C) 12041 (D) 1034

Q.30 What is the output printed by the following C code ?

```
# include <stdio.h>
int  main ( )  {
char a [ 6 ] = "world" ;
int i,  j ,  ;
for ( i = 0, j = 5; i < j ; a [ i ++ ] = a [ j- - ] );
printf ( " % s\n", a );  }
```

(A) dlrow (B) Null string (C) dlrld (D) worow

(21) ⇒

```
void main()
{
    int a[2][3][2] = {1,2,3,4,5,6,7,8,9,10,
                      11,12};
    printf("%u %u %u %u", a, *a, **a, ***a);
    printf("%u %u %u %u", a+1, *a+1, **a+1, ***a+1);
}
```

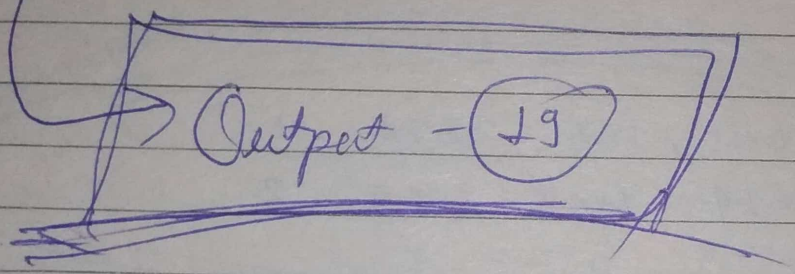
Base Address = 1000, int → 4 bytes.

1000, 1000, 1000, 1
1024, 1008, 1004, 2

(22) ⇒

```
#include<stdio.h>
int f(int x, int *py, int **ppz)
{
    int y, z;
    **ppz += 1;
    z = **ppz;
    *py += 2;
    y = *py;
    x += 3;
    return x+y+z;
}
```

```
void main()
{
    int c, *b, **a;
    c = 4;
    b = &c;
    a = &b;
    printf("%d", f(c, b, a));
    getchar();
}
```



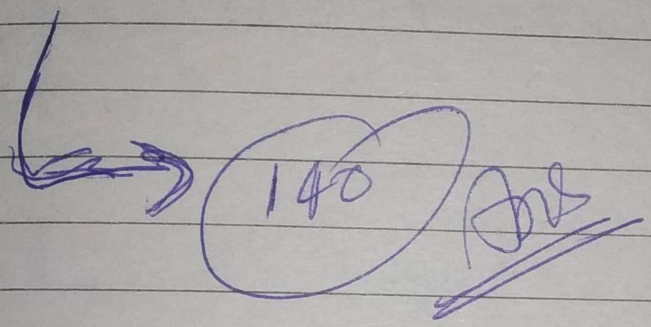
23 ⇒

```
int (*f)(int *);
```

A pointer to a f^n that take an integer pointer as argument and return an integer.

24

```
#include<stdio.h>
int main()
{
    static int a[] = {10, 20, 30, 40, 50};
    static int *p[] = {a, a+3, a+4, a+1, a+2};
    int **ptr = p;
    ptr++;
    printf("%d%d", ptr-p, **ptr);
}
```



25

```
int main()
{   int array[] = {3, 5, 1, 4, 6, 2};
    int done = 0;
    int i;
    while(done == 0)
    {   done = 1;
        for(i=0; i<=4; i++)
        {   if(array[i] < array[i+1])
            {   swap(&array[i], &array[i+1]);
                done = 0;
            }
        }
```

```
for (i=5; i>=1; i--)
{
    if (array[i] > array[i-1])
    {
        swap(&array[i], &array[i-1]);
        done = 0;
    }
}
printf("%d", array[3]);
```

→ Output - 3

26 ⇒

```
void swap(int *px, int *py)
{
    *px = *px - *py;
    *py = *px + *py;
    *px = *py - px;
}
```

52 May generate a segmentation fault at run-time depending on the argument passed

54 Implement the swap procedure correctly for some but not all valid i/p pointers.

27)

```
#include <stdio.h>
int main()
{
    int a[4][5] = {{1,2,3,8,5},
                   {6,7,8,9,10},
                   {11,12,13,14,15},
                   {16,17,18,19,20}};
    printf("%d" * (*a + **a + 2) + 3));
    return (0);
}
```

Output = 7 Ans

28)

```
int f(int *p, int n)
{
    if (n <= 1) return 0;
    else return max(f(p+1, n-1), p[0] - p[1]);
}

int main()
{
    int a[] = {3, 5, 2, 6, 4};
    printf("%d", f(4, 5));
}
```

Output - 3

29 ⇒

```
#include<stdio.h>
int main()
{
  char s1[7]="1234", *p;
  p= s1+2;
  *p='0';
  printf("%s", s1);
}
```

→ Output = 1204 Ans

30 ⇒

```
#include<stdio.h>
int main()
{
  char a[6]="world";
  int i, j;
  for(i=0,j=5; i<j; a[i++]=a[j--]);
  printf("%s", a);
}
```

→ Output → NULL Ans