Q.82

```
#include <stdio.h>
static char c;
static int i;
static float f;
int main(){
printf("%d %d %f ",c,i,f);
return 0;
}
```

The output is _____

(A) 0 0 0.000000

(B) 0 0 0

(C) Compilation Error

(D) None

Q.84 Which of the following will create an error

(A)

```
#include <stdio.h>
static int i=2;
i=55;
int main(){
printf("%d",i);
return 0;
}
```

Q.86 What will be output of following c code?

```
#include<stdio.h>
int main(){
int a=0;
{
int a=10;
printf("%d",a);
a++;
{
a=20;
}
{
printf(" %d",a);
int a=30; {a++;}
printf(" %d",a++);
}
printf(" %d",a++);
}
printf(" %d",a);
return 0;
}
```

10

20

31

20

0

Output :-

10203L200

Q.87 The output of the code :

```
#include<stdio.h>
int main(){
int i;
for(i=0;i<6;i++){
int a=20;
printf("%d",a);
a++;
}
return 0;
}
```

is _____

20202020220 → output

Q.88 Which of the following is/are true regarding register variable :

(A) A register variable execute faster than other variables.

(B) Default initial value of register variable is garbage.

(C) Scope and visibility of register variable is block.

(D) The register modifier tells the compiler to do its best to keep the variable in a register if at all possible. Otherwise it is stored on the stack.

(E) All of these

Q.89 Which of the following is/are true regarding static variable:

(A) static variables are stored in a data segment.

(B) static variables have visibility from the point of declaration to the end of the enclosing scope.

(C) A same static variable can be declared many times but we can initialize at only one time.

(D) We cannot write any assignment statement globally .

(E) All of these

Q.90 Which of the following is/are true regarding static variable:

(A) A static variable initializes only one time in whole program.

(B) If we declared static variable locally then its visibility will within a block where it has declared.

(C) If declared a static variable or function globally then its visibility will only the file in which it has declared not in the other files.

(D) A static variable has internal linkage.

(E) All of these

Q.91 What will be the output of the C program?

```
#include<stdio.h>
static int i=2;
extern int i;
int main()
{
    printf("%d",i);
    return 0;
}
```

(A) 0 (B) 2 (C) Compilation Error (D) Runtime Error

Q.92 What will be the output of the C program?

```
#include<stdio.h>
static int i=2;
extern int i=4;
int main()
{
    printf("%d",i);
    return 0;
}
```

(A) 2 (B) 4 (C) 0 (D) Compilation Error

Q.93

```
#include <stdio.h>
int * build_array();
int main() {
int *a,k;
```

```
    a = build_array();
    for (k = 0; k < 5; k++)
    printf("%d ", a[k]);
    return 0;
}
    int * build_array()
{
    static int arr[5]={1,2,3,4,5};
    return arr;
}
```

The output is :

(A) Compilation Error (B) Runtime Error

(C) 1 2 3 4 5 (D) None

Q.94

```
#include <stdio.h>
    int * build_array();
    int main() {
    int *a,k;
    a = build_array();
    for (k = 0; k < 5; k++)
    printf("%d ", a[k]);
    return 0;
}
    int * build_array()
{
    int arr[5]={1,2,3,4,5};
     return arr;
}
```

The output is :

(A) 1 2 3 4 5 (B) Segmentation Error

(C) Null pointer Error (D) None

Q.95

```
#include <stdio.h>
    char *p1[] = {"Gurupal", "Sir", "Math", "King"};
    char **p2[] = {p1+3, p1+2, p1+1, p1};
    char ***ptr = p2;
    int main()
{
    printf("%s ", **++ptr);
```

→ Math

```
    printf("%s ", *--*++ptr+3);
    printf("%s ", *ptr[-2]+3);
    printf("%s ", ptr[-1][-1]+1);
    return 0;
}
```

(A) Math upal g ir
(B) Sir pal g ir
(C) Math pal g ir
(D) GarbageValue upal g ir

Q.96

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i;
    int *ptr = (int *) malloc(5 * sizeof(int));
    for (i=0; i<5; i++)
    *(ptr + i) = i;
printf("%d ", *ptr++);
printf("%d ", (*ptr)++);
printf("%d ", *ptr);
printf("%d ", *++ptr);
printf("%d ", ++*ptr);
}
```

(A) Compiler Error
(B) 0 1 2 2 3
(C) 0 1 2 3 4
(D) 1 2 3 4 5

Q.97

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

(A) Compiler Error
(B) 3 1
(C) 2 2
(D) 1 1