Type: MCQ

Q1. What is the output of following C program? ? (0.5)

```
#include< stdio.h>
#include< string.h>
int main()
{
static char str1[] = "dills";
static char str2[20];
static char str3[] = "Daffo";
int i;
i = strcmp(strcat(str3, strcpy(str2, str1)), "Daffodills");
printf("%d", i);
return 0;
}
```

1) ** 0   2) 1   3) -1   4) Compile error

Q2. What will be the output of the following C program? (0.5)

```
# include <stdio.h>
#include<string.h>
int main()
{
char p[20];
char s[10] = "string";
int length = strlen(s);
int i;
for (i = 0; i < length; i++)
p[i] = s[length  - i];
printf("%s", p);
}
```

1) gnirts 2) gnirt 3) string 4) ** No output is printed

Q3. What will be the output of the following C program? (0.5)

```
#include<stdio.h>
int main()
{
 int fun[2][3]={{1,2,3},{4,5,6}}, i=3-2*1/1, j=3*1-3/1;
 fun[i][j]=fun[j][i];
 printf("%d\t %d",fun[i/1][j*1],fun[j+1/2*1][i+1/3*1]);
 return 0;
 }
```

1) 2 4   2) ** 2 2   3) 2 6   4) 2 5

Q4. Identify true or false. (0.5)

i. The global variable(s) can be accessed in the main() but not in other functions of the same program.

ii. In C language, there can be functions with no arguments but a return value.

1) i-True, ii-True 2) i- True, ii-False 3) ** i-False, ii-True
4) i- False, ii-False

Q5. Which among the following is an invalid function prototype? (0.5)
1) int funct (int, int); 2) int funct (int, char); 3) int funct (void); 4) ** int funct (int, void, int);

Q6. Identify the value for variable k and l in the C program given below such that the output is 0 0 0 2 0 3? (0.5)

```
int main()
{
int a[][3] ={{0}, {2, 0, 3 } } , i , j , k , l;
for(i=0;i< k;i++)
for(j=0;j< l;j++)
printf("%d ",a[i][j]);
return 0;
}
```

1) k=3, l=2 2) k=1, l=6 3) ** k=2, l=3 4) k=3, l=3

Q7. What is the output of the following code? (0.5)

```
#include< stdio.h>
void f(int n)
{
if (n > 0)
{
printf("%d",n%10);
f(n / 10);
}
}
int main()
{
f(4567);
}
```

1) ** 7654 2) 0567 3) 7650 4) 4567

Q8. What is the output of the following C code? (0.5)

```
#include< stdio.h>
int ABC(int inum)
{
if( inum==0)
return(1);
return(inum* ABC(inum-1));
}
int main()
{
```

```
int c;
c=ABC(5);
printf("%d",c);
}
```

1) 0 2) ** 120 3) 1 4) 25 }

Q9. What is the output of the following error free code:

```
#include < stdio.h>
int main()
{
int i = 97, *p = &i;
printf("%d ", *p);
}
void too(int *p)
{
int j = 2;
p = &j;
printf("%d ", *p);
}
```

1) 2 2 2) ** 2 97 3) 97 2 4) 97 97

Q10. What is the output of the following C code? (0.5)

```
#include < stdio.h>
int main()
{
int *ptr;
int x;
ptr = &x;
*ptr = 0;
printf(" x = %d\t", x);
printf(" *ptr = %d\t", *ptr);
*ptr += 5;
printf(" x = %d\t", x);
printf(" *ptr = %d\t", *ptr);
(*ptr)++;
printf(" x = %d\t", x);
printf(" *ptr = %d\t", *ptr);
return 0;
}
```

1) ** x =0 *ptr=0 x=5 *ptr=5 x=6 *ptr=6

2) x=garbage value *ptr=0 x=garbage value *ptr=5 x=garbage value *ptr=6

3) x =0 *ptr=0 x=5 *ptr=5 x=garbage value *ptr=garbage value

4) x =0 *ptr=0 x =0 *ptr=0 x =0 *ptr=0

Type: DES

Q11 Explain with example different ways of initializing strings. Write a C program to read a string and remove blank spaces from the string and display the modified string. (3)

Soluation:

Different ways of initializing strings:

(a) Individual Characters are written inside Single Quotes , Separated by comma to form a list of characters. Complete list is wrapped inside **Pair of Curly braces.** NULL Character should be written in the list.
Example: **char myWord [ ] = { 'H', 'e', 'l', 'l', 'o', '\0' }; 0.5M**

(b) In this method we are directly assigning String to variable by writing text in double quotes. In this type of initialization, we don't need to put NULL or **Ending / Terminating character** at the end of string. It is appended automatically by the compiler.

Example: char myWord [ ] = "Hello"; 0.5M

[String Input-0.5M, Looping: 0.5M Logic 1M]

```
#include <stdio.h>
int main()
{
  char text[100], blank[100];
  int c = 0, d = 0;
  printf("Enter some text\n");
  gets(text);
  while (text[c] != '\0')
  {
    if (!(text[c] == ' ' && text[c+1] != ' ')) {
      blank[d] = text[c];
      d++;
    }
    c++;
  }
  blank[d] = '\0';
  printf("Text after removing blanks\n%s\n", blank);
  return 0;
}
```

Q12. Write a C program that reads MXN 2D array to check and display if the matrix is "identical" or "non-identical".
[Hint: If the product of primary diagonal elements is same as the product of secondary diagonal elements it is "identical" matrix otherwise it is "non-identical"]. (2)

Soluation.

```
#include<stdio.h>


int main()

{

   int arr[10][10],rows,cols,i,j, prim=1,sec=1;


   printf("enter the size of matrix (row,cols)");

   scanf("%d%d",&rows,&cols);


   for(i=0;i<rows;i++)

   {

      for(j=0;j<cols;j++)

      {

         scanf("%d",&arr[i][j]);

      }

   }

   for(i=0;i<rows;i++)

   {

         prim=prim*arr[i][i];

         sec=sec*arr[i][cols-i-1];

   }

   if(prim==sec)

      printf("\n identical");

   else

      printf("\n non-identical");

   return 0;

}
```

Read matrix: 0.5 M+ find primary product and secondary product 0.5M + 0.5M+ display identical/ non-identical 0.5M

Q13. Write a function void sort (int m, int n, int x[][10]) to sort the elements row wise in ascending order. Write a C program which reads a 2D array and display the sorted array in main(). (3)

Soluation:

```
#include <stdio.h>
void sort_mat(int ma[][10],int m,int n)
{
   int i,k,j,a;
  for (i=0;i<m;++i) {                                                    1/2M
                for (j=0;j<n;++j) {
                        for (k=(j+1);k<n;++k) {
                                if (ma[i][j] > ma[i][k]) {
                                a = ma[i][j];
                                ma[i][j] = ma[i][k];
                                ma[i][k] = a;

                                }
                        }
                }
        }
}                                                                          1M

void main () {

        int ma[10][10],i,j,k,a,m,n;
        printf ("Enter the order of the matrix \n");
        scanf ("%d %d", &m,&n);
        printf ("Enter co-efficients of the matrix \n");
        for (i=0;i<m;++i) {
                for (j=0;j<n;++j) {
                        scanf ("%d",&ma[i][j]);
                }
        }                                                              1/2M
printf ("After arranging rows in ascending order\n");
sort_mat(ma,m,n);                                                      1/2M
        for (i=0;i<m;++i) {
                for (j=0;j<n;++j) {
                printf (" %d",ma[i][j]);
                }

                printf ("\n");} }                                      1/2M
```

Q14. Write a C program that reads 1D array and displays the total number of perfect square numbers in the array in main() using a function “int CheckPerfect(int)” to check whether the given number is perfect square or not. (2)

Soluation:

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

int CheckPerfect(int N)                                                     0.5M
{
   float sqrtnum=sqrt(N);
   int num1=sqrtnum, num2=sqrtnum+1;
   if((num1*num1==N)||(num2*num2==N))                                        0.5M
       return 1;
   else
      return 0;
}

int main()
{
   int N,arr[10],i,count=0;
   printf("Enter the number of elements to be inserted\n");
   scanf("%d",&N);

   printf("Enter the elements to be inserted:\n");                           0.5M
   for(i=0;i<N;i++)
   {
      scanf("%d",&arr[i]);
   }
   for(i=0;i<N;i++)
   {
      if(CheckPerfect(arr[i]))                                               0.5M
         count++;
   }
   printf("\n There are %d perfect square numbers", count);
   return 0;
}
```