SAP ID - 60004200132
Name : Ayush Jain
Div - I
Branch - Computer Engineering

23/08/2021 Term Test - 2

Computer Programming

Q. 1

→ A function that calls itself is called as a recursive function. A recursive function must definitely have a condition that exists from calling the function again. Hence there must be a condition that calls the function itself if that condition is true. If the condition is false then it will exit from the loop of calling itself again.

Program

```
# include <stdio.h>
# include <conio.h>
void main ()
{
```

Program:

```
# include <conio.h>
# include <stdio.h>
int find_factorial (int);

int main ()
{
  int num, fact;
  Printf ("\n Enter any integer number: ");
  scanf (" %d", &num);
  fact = find_factorial (num);
  Printf ("\n Factorial of %d is : %d", num, fact);

  return 0;
}
Int  find_factorial (int n)
{
   if (n==0)
      return(1);
   else
      return (n* find_factorial(n-1));
}
```

Output:

```
Enter any integer number : 4
Factorial of 4 is : 24
```

Q. 2) Program

```
#include <stdio.h>
int binarySearch (int arr[], int low, int high, int key)
  {
    if (high < low)
        return -1;
   int mid = (low + high)/2 ; low + (high - low)/2 ;
if (key == arr[mid])
      return mid;
if (key > arr [mid])
    return binarySearch (arr, bw, (mid -1), key);
}

int main()
{
  int arr [10];
  int n, c, key;
  Printf (" Enter size of array: ");
  scanf (" Enter %d", &c);
  Printf (" Enter a sorted array : ");
  for (int i=0; i<c; i++) {
   scanf ("%d", &arr [i])
   n = sizeof(arr) / sizeof (arr[0]);
  }
  print f (" EnHr key : ");
  scanf ("%d", &key);
 printf (" Index : %d \n", binary Search (arr, 0, n-1, key));
}
```

Output:

```
Enter size of array: 5
Enter a sorted array: 1 2 3 4 5
Enter key: 2
Index: 1
```

Q. 3) Program:

```
#include<stdio.h>
void change (int* num)
{
Printf "Before adding value inside function num = %
```

Q. 3. Program :

```
# include <stdio.h>
void swap (int* , int* );
int main ( )
{
 int a = 20 ;
 int b = 40 ;
printf (" Before swapping the values a= %d , b = %d \n ", a, b);
 swap (&a, &b);
printf (" After swapping the values  a= %d, b= %d \n", a, b);
}
void swap ( int*a , int* b )
{
int temp;
tem = * a ;
* a = * b ;
* b = temp ;
printf (" After swapping values in function a= %d , b= %d \n" a, b);
}
```

Output :

```
Before swapping the values  a = 20 , b = 40
After swapping the values in function  a = 40 , b = 20
After swapping the values  a = 40 , b = 20
```