Assignment - 10 A Job Ready Bootcamp in C++, DSA and IOT MySirG
Functions in C Language

1. Write a function to calculate the area of a circle. (TSRS)

Program -

```
#include <stdio.h>

float areaOfCircle(int);

int main()
{
   printf("Area of circle is %f",areaOfCircle(5));
   return 0;
}

float areaOfCircle(int r)
{
  return 3.14159 * r * r;
}
```

Output -

Area of circle is 78.539749

---

2. Write a function to calculate simple interest. (TSRS)

Program -

```
#include <stdio.h>

float SI(int,int,int);

int main()
{
   printf("Simple Interest = %f",SI(5000,3,4));
   return 0;
}

float SI(int p , int r , int t)
{
    return (p * r * t) / 100;
}
```

Output -

Simple Interest = 600.000000

---

3. Write a function to check whether a given number is even or odd. Return 1 if the number is even, otherwise return 0. (TSRS)

Program -

```
#include <stdio.h>

int checkEven(int);

int main()
{
  int num;

  printf("Enter a number: ");
  scanf("%d",&num);

  if(checkEven(num))
    printf("Even");
  else
    printf("Not an even number");

  return 0;
}

int checkEven(int n)
{
    if(n % 2)
        return 0;
    else
        return 1;
}
```

Output -

Enter a number: 8
Even

---

4. Write a function to print first N natural numbers (TSRN)

Program -

```
#include <stdio.h>

void printN(int);

int main()
{
  int num;

  printf("Enter a number: ");
  scanf("%d",&num);
```

```
  printf("First %d natural numbers are\n",num);
  printN(num);

  return 0;
}

void printN(int n)
{
   int i;
   for(i = 1 ; i <= n ; i++)
   {
       printf("%d ",i);
   }
}
```

Output -
Enter a number: 8
First 8 natural numbers are
1 2 3 4 5 6 7 8

---

5. Write a function to print first N odd natural numbers. (TSRN)

Program -

```
#include <stdio.h>

void printOdd(int);

int main()
{
  int num;

  printf("Enter a number: ");
  scanf("%d",&num);

  printf("First %d Odd natural numbers are\n",num);
  printOdd(num);

  return 0;
}

void printOdd(int n)
{
   int i;
   for(i = 1 ; i <= n ; i++)
   {
       printf("%d ",2 * i - 1);
   }
}
```

Output -
Enter a number: 10
First 10 Odd natural numbers are
1 3 5 7 9 11 13 15 17 19

---

6. Write a function to calculate the factorial of a number. (TSRS)

Program -
```
#include <stdio.h>

int fact(int);

int main()
{
  int num;

  printf("Enter a number: ");
  scanf("%d",&num);

  printf("Factorial of %d is %d",num,fact(num));

  return 0;
}

int fact(int n)
{
  int f = 1;

  while(n)
  {
      f *= n;
      n--;
  }
  return f;
}
```

Output -
Enter a number: 7
Factorial of 7 is 5040

---

7. Write a function to calculate the number of combinations one can make from n items and r selected at a time. (TSRS)

Program -
```
#include <stdio.h>

int fact(int);
```

```
int combi(int,int);

int main()
{
  int n , r;
  printf("Enter total number of items: ");
  scanf("%d",&n);
  printf("Enter number of items to be selected from %d items:
",n);
  scanf("%d",&r);

  if(r > n || r < 0)
  {
      printf("Combinations not possible!");
      printf("\nEither number of items selected is more than %d or
less than 0",n);
      return 0;
  }

  printf("Number of combinations possible = %d",combi(n,r));

  return 0;
}

int fact(int n)
{
  int f = 1;

  while(n)
  {
      f *= n;
      n--;
  }
  return f;
}

int combi(int n , int r)
{
    return fact(n)/(fact(n - r) * fact(r));
}
```

Output -

Enter total number of items: 5
Enter number of items to be selected from 5 items: 2
Number of combinations possible = 10

8. Write a function to calculate the number of arrangements one can make from n items and r selected at a time. (TSRS)

Program -

```
#include <stdio.h>

int fact(int);
int arrange(int,int);

int main()
{
  int n , r;
  printf("Enter total number of items: ");
  scanf("%d",&n);
  printf("Enter number of items to be selected from %d items:
",n);
  scanf("%d",&r);

  if(r > n || r < 0)
  {
      printf("Arrangement not possible!");
      printf("\nEither number of items selected is more than %d or
less than 0",n);
      return 0;
  }

  printf("Number of arrangements of %d things selected %d at a
time: %d",n,r,arrange(n,r));

  return 0;
}

int fact(int n)
{
  int f = 1;

  while(n)
  {
      f *= n;
      n--;
  }
  return f;
}

int arrange(int n , int r)
{
    return fact(n)/fact(n - r);
}
```

Output -

Enter total number of items: 6
Enter number of items to be selected from 6 items: 3
Number of arrangements of 6 things selected 3 at a time: 120

---

9. Write a function to check whether a given number contains a given digit or not. (TSRS)

Program -

```
#include<stdio.h>

int checkDigit(int,int);

int main()
{
    int num , digit;

    printf("Enter a number: ");
    scanf("%d",&num);
    printf("Enter a digit: ");
    scanf("%d",&digit);

    if(checkDigit(num,digit))
        printf("Digit %d is present in %d",digit,num);
    else
        printf("Digit %d is not present in %d",digit,num);

    return 0;
}

int checkDigit(int n , int d)
{
    int rem , flag = 0;
    while(n)
    {
        rem = n % 10;
        if(rem == d)
        {
            flag = 1;
            break;
        }
        n /= 10;
    }
    if(flag)
        return 1;
    else
        return 0;
}
```

Output -
Enter a number: 1963
Enter a digit: 6
Digit 6 is present in 1963

---

10. Write a function to print all prime factors of a given number. For example, if the number is 36 then your result should be 2, 2, 3, 3. (TSRN)

Program -

```
#include<stdio.h>

void primeFactors(int);

int main()
{
    int num;
    printf("Enter a number: ");
    scanf("%d",&num);

    primeFactors(num);

    return 0;
}

void primeFactors(int n)
{
    int i;
    for(i = 2 ; n > 1 ; i++)
    {
        while(n % i == 0)
        {
            printf("%d ",i);
            n = n / i;
        }
    }
}
```

Output -
Enter a number: 16
2 2 2 2
