Assignment – 10

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

```c
#include<stdio.h>
float area(int r)
{
    float ar;
    ar = 3.141592*r*r;
    return ar;
}
int main()
{
    int r;
    printf("Enter radius of circle :");
    scanf("%d",&r);
    printf("%f",area(r));
    return 0;
}
```

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

```c
#include<stdio.h>
float SimpleInterest(float p,float r,float t)
{
    float si;
    si = (p*r*t)/100 ;
    return si;
}
int main()
{
    float p,r,t;
    printf("Enter Principal Amount :");
    scanf("%f",&p);
    printf("Enter Rate of Interest in %% :");
    scanf("%f",&r);
    printf("Enter Time :");
    scanf("%f",&t);
    printf("%f",SimpleInterest(p,r,t));
    return 0;
}
```

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)

```c
#include<stdio.h>
int oddeven(int n)
{
    if(n%2==0)
        return 1;
    else
        return 0;
}
int main()
{
    int n;
```

```
    printf("Enter Number :");
    scanf("%d",&n);
    if(oddeven(n))
    {
        printf("Given number is even");
    }
    else
    {
        printf("Given number is odd");
    }
    return 0;
}
```

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

```
#include<stdio.h>
void printnaturalN(int n)
{
    int i;
    for(i=1;i<=n;i++)
    {
        printf("%d ",i);
    }
}
int main()
{
    int n;
    printf("Enter value of N :");
    scanf("%d",&n);
    printnaturalN(n);
    return 0;
}
```

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

```
#include<stdio.h>
void oddnaturalN(int n)
{
    int i;
    for(i=1;i<=n*2;i++)
    {
        if(i%2==1)
            printf("%d ",i);
    }
}
int main()
{
    int n;
    printf("Enter value of N :");
    scanf("%d",&n);
    oddnaturalN(n);
    return 0;
}
```

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

```c
#include<stdio.h>
int fact(int n)
{
    int i,result=1;
    for(i=1;i<=n;i++)
    {
        result = result * i ;
    }
    return result;
}
int main()
{
    int n;
    printf("Enter value of N :");
    scanf("%d",&n);
    printf("Factorial of %d is %d",n,fact(n));
    return 0;
}
```

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

```c
#include<stdio.h>
int fact(int n)
{
    int i,result=1;
    for(i=1;i<=n;i++)
    {
        result = result * i ;
    }
    return result;
}
int nCr(int n,int r)
{
    int result;
    result = (fact(n)/(fact(r)*fact(n-r)));
    return result;
}
int main()
{
    int n,r;
    printf("Enter number of items :");
    scanf("%d",&n);
    printf("Enter number of items to be selected :");
    scanf("%d",&r);
    printf("Total number of combination possible : %d",nCr(n,r));
    return 0;
}
```

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

```c
#include<stdio.h>
int fact(int n)
{
    int i,result=1;
    for(i=1;i<=n;i++)
    {
        result = result * i ;
    }
    return result;
}
int nPr(int n,int r)
{
    int result;
    result = (fact(n)/(fact(n-r)));
    return result;
}
int main()
{
    int n,r;
    printf("Enter number of items :");
    scanf("%d",&n);
    printf("Enter number of items to be selected :");
    scanf("%d",&r);
    printf("Total number of arrangements possible : %d",nPr(n,r));
    return 0;
}
```

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

```c
#include<stdio.h>
int find(int n,int d)
{
    int mod;
    while (n>0)
    {
       mod = n%10;
       n = n/10;
       if(mod == d)
            return 1;
    }
    return 0;
}
int main()
{
    int n,d;
    printf("Enter Number :");
    scanf("%d",&n);
    printf("Enter Digit :");
    scanf("%d",&d);
    if(find(n,d))
    {
        printf("%d contains digit %d",n,d);
    }
    else
    {
        printf("%d does not contains digit %d",n,d);
```

```
    }
    return 0;
}
```

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)

```
#include<stdio.h>
void primefact(int n)
{
    int i=2;
    printf("Prime factor of %d is ",n);
    while (n>1)
    {
        if(n%i==0)
        {
            printf("%d ",i);
            n = n/i;
        }
        else if (n==1 || n==0)
        {
            break;
        }
        else
        {
            i++;
        }
    }
}
int main()
{
    int n;
    printf("Enter Number :");
    scanf("%d",&n);
    primefact(n);
    return 0;
}
```