

Assignment - 10

Functions in C Language

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

```
#include<stdio.h>
float area(float);
int main()
{
    float r;
    printf("\nEnter Radius of circle: ");
    scanf("%f", &r);
    printf("Area of Circle is: %.2f", area(r));
    return 0;
}
float area(float a)
{
    return 3.14*a*a;
}
```

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

```
#include<stdio.h>
float si(float, float, float);
int main()
{
    float P, T, R;
    printf("\nEnter Principle, Time and Rate: ");
    scanf("%f %f %f", &P, &T, &R);
    printf("Simple Interest is: %.2f", si(P,T,R));
    return 0;
}
float si(float P, float T, float R)
{
    return (P * T * R) / 100;
}
```

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)

```
#include<stdio.h>
int check(int);
int main()
{
```

```

int n, num;
printf("Enter number: ");
scanf("%d", &n);
num = check(n);
if (num==1)
{
    printf("\nEven number");
}
else
{
    printf("\nOdd number");
}
}

int check(int n)
{
    if (n%2==0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

```

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

```

#include<stdio.h>
void natural(int);
int main()
{
    int n;
    printf("\nEnter number: ");
    scanf("%d", &n);
    natural(n);
}

void natural(int n)
{
    int i;
    for ( i = 1; i <= n; i++)
    {
        printf("%d\t", i);
    }
}

```

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

```
#include<stdio.h>
void odd(int);
int main()
{
    int n;
    printf("Enter number: ");
    scanf("%d", &n);
    odd(n);
}

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

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

```
#include<stdio.h>
int fact(int);
int main()
{
    int n;
    printf("Enter number: ");
    scanf("%d", &n);
    printf("%d", fact(n));
}

int fact(int n)
{
    int i, fact = 1;
    for ( i = 1; i <= n; i++)
    {
        fact = fact * i;
    }
    return fact;
}
```

7. Write a function to calculate the number of combinations one can make from n items

and r selected at a time. (TSRS)

```

#include<stdio.h>
int fact(int);
int combi(int, int);
int main()
{
    int n, r;
    printf("\nEnter n items and r selected at a time : ");
    scanf("%d %d", &n, &r);
    printf("\nthe number of combinations one can make is %d", combi(n, r));
}
int combi(int n, int r)
{
    int a = fact(n);
    int b = fact(r);
    int combi = a / (b * fact(n - r));
    return combi;
}
int fact(int num)
{
    int i, fact = 1;
    for (i = 0; i < num; i++)
    {
        fact = fact * (num - i);
    }
    return fact;
}

```

8. Write a function to calculate the number of arrangements one can make from n items

and r selected at a time. (TSRS)

```

#include<stdio.h>
int fact(int);
int permu(int, int);
int main()
{
    int n, r;
    printf("\nEnter n items and r selected at a time : ");
    scanf("%d %d", &n, &r);
    printf("\nthe number of arrangements one can make is %d", permu(n, r));
}
int permu(int n, int r)
{
    int permu = fact(n) / (fact(n - r));
    return permu;
}

```

```

}

int fact(int num)
{
    int i, fact = 1;
    for ( i = 0; i < num; i++)
    {
        fact = fact * (num - i);
    }
    return fact;
}

```

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

```

#include<stdio.h>
int num(int, int);
int main()
{
    int n = 2468, d;
    printf("\nEnter a digit to check whether its present in a number or not:");
    scanf("%d", &d);
    if (num(n, d))
    {
        printf("\nDigit is present.");
    }
    else
    {
        printf("\nDigit is not present");
    }
}

int num(int n, int d)
{
    while (n)
    {
        if (n%10==d)
        {
            return 1;
        }
        else
        {
            n = n/10;
        }
    }
    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);
int main()
{
    int n = 36;
    primefact(n);
}
void primefact(int n)
{
    int i = 2;
    while (n)
    {
        if (n%i==0)
        {
            n = n/i;
            printf("%d\t", i);
        }
        else
        {
            i++;
        }
    }
}
```