**Assignment-10 Solution Name: Om Pant**

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

Ans-

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

*#include*<stdio.h>

float areaOfcircle(float r){

*return* 3.14\*r\*r;

}

int main(){

    float radius,area;

    printf("Enter radius of circle\n");

    scanf("%f",&radius);

    area = areaOfcircle(radius);

    printf("The Area of circle having radius %.2f is %.2f\n",radius,area);

*return* 0;

}

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

Ans –

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

*#include*<stdio.h>

int simpleInt(int p, int r, int t){

    int intrest;

    intrest = (p\*r\*t)/100;

*return* intrest;

}

int main(){

    int p,r,t,si;

    printf("Enter Principle, rate and time \n");

    scanf("%d%d%d",&p,&r,&t);

    si = simpleInt(p,r,t);

    printf("Simple Interest is :%d\n",si);

*return* 0;

}

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

Ans-

*#include*<stdio.h>

int checkNo(int n);

int main(){

    int num,result;

    printf("Enter a number\n");

    scanf("%d",&num);

    result = checkNo(num);

*if*(result)

        printf("Even number\n");

*else*

        printf("Odd number\n");

*return* 0;

}

int checkNo(int n){

*if*(n%2 == 0)

*return* 1;

*else*

*return* 0;

}

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

Ans-

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

*#include*<stdio.h>

void naturalNum(int n){

*for*(int i=1;i<=n;i++){

        printf("%d\t",i);

    }

}

int main(){

    int num;

    printf("Enter the value of n\n");

    scanf("%d",&num);

    naturalNum(num);

*return* 0;

}

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

Ans –

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

*#include*<stdio.h>

void oddNaturalNum(int n){

*for*(int i=1;i<=n;i++){

        printf("%d\t",2\*i-1);

    }

}

int main(){

    int num;

    printf("Enter the value of n\n");

    scanf("%d",&num);

    oddNaturalNum(num);

*return* 0;

}

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

Ans-

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

*#include*<stdio.h>

int factorial(int n){

    int temp = 1;

*for*(int i=n;i>=1;i--){

        temp = temp \* i;

    }

*return* temp;

}

int main(){

    int number,fact;

    printf("Enter a number\n");

    scanf("%d",&number);

    fact = factorial(number);

    printf("Factorial of %d is %d\n",number,fact);

*return* 0;

}

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

Ans-

*// 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>

long long int factorial(long long int n){

    long long int temp = 1;

*for*(long long int i=n;i>=1;i--){

        temp = temp \* i;

    }

*return* temp;

}

long long int combinations(long long int n, long long int r){

    long long int factN,factR,factDiff,combi;

    factN = factorial(n);

    factR = factorial(r);

    factDiff = factorial(n-r);

    combi = factN / (factR \* factDiff);

*return* combi;

}

int main(){

    long long int n,r,result;

    printf("Enter No.of items and no. of Selected items\n");

    scanf("%lld%lld",&n,&r);

    result = combinations(n ,r);

    printf("%lld Combinations Possible\n",result);

}

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

Ans –

*// 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>

long long int factorial(long long int n){

    long long int temp = 1;

*for*(long long int i=n;i>=1;i--){

        temp = temp \* i;

    }

*return* temp;

}

long long int permutations(long long int n, long long int r){

    long long int factN,factDiff,combi;

    factN = factorial(n);

    factDiff = factorial(n-r);

    combi = factN / factDiff;

*return* combi;

}

int main(){

    long long int n,r,result;

    printf("Enter No.of items and no. of Selected items\n");

    scanf("%lld%lld",&n,&r);

    result = permutations(n ,r);

    printf("%lld Arrangments/Permutations Possible\n",result);

}

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

Ans-

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

*#include*<stdio.h>

int checkDig(int n,int d){

    int x;

*while*(n!=0){

            x = n%10;

            n = n/10;

*if*(x == d){

*return* 1;

            }

    }

*return* 0;

}

int main(){

    int num,dig,result;

    printf("Enter a number\n");

    scanf("%d",&num);

    printf("Enter a Digit\n");

    scanf("%d",&dig);

    result = checkDig(num , dig);

*if*(result)

        printf("Number Contains the Given Digit\n");

*else*

        printf("Number doesn't Contains the Given Digit\n");

*return* 0;

}

1. 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)

Ans –

*// 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>

*# include* <math.h>

void primeFactors(int n)

{

*while* (n%2 == 0){

        printf("%d ", 2);

        n = n/2;

    }

*for* (int i = 3; i <= sqrt(n); i = i+2){

*while* (n%i == 0){

            printf("%d ", i);

            n = n/i;

        }

    }

*if* (n > 2)

        printf ("%d ", n);

}

int main()

{

    int n;

    printf("Enter a Number to find its prime Factors\n\n");

    scanf("%d",&n);

    primeFactors(n);

*return* 0;

}