**Functions in C Language**

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

#include<stdio.h>

float area(int);

int main ()

{

int r;

float a;

printf("Enter a radius.");

scanf("%d",&r);

a=area(r);

printf("area of Circle is %.2f",a);

return 0;

}

float area(int x)

{

float a1=3.14\*x\*x;

return a1;

}

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

#include<stdio.h>

float SI(int,int,int);

int main ()

{

int amount,rate,time;

float S;

printf("Enter a amount,rate and time.");

scanf("%d%d%d",&amount,&rate,&time);

S=SI(amount,rate,time);

printf("Simple Interest of %dRs is %.2fRs",amount,S);

return 0;

}

float SI(int am,int ra,int ti)

{

float si=(am\*ra\*ti)/100;

return si;

}

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)***

#include<stdio.h>

int even\_odd(int);

int main ()

{

int num,ans;

printf("enter a number:");

scanf("%d",&num);

ans=even\_odd(num);

printf("returned number 1 or 0 when even or odd respectively.\n");

printf("\n%d",ans);

return 0;

}

int even\_odd(int num)

{

if(num%2==0)

return 1;

else

return 0;

}

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

#include<stdio.h>

void natural\_no(int);

int main ()

{

int N;

printf("enter N number:");

scanf("%d",&N);

natural\_no(N);

return 0;

}

void natural\_no(int num)

{

int i;

for(i=1; i<=num; i++)

printf("%d\n",i);

}

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

#include<stdio.h>

void firstn\_odd(int);

int main ()

{

int N;

printf("enter N number:");

scanf("%d",&N);

firstn\_odd(N);

return 0;

}

void firstn\_odd(int num)

{

int i,j=1;

for(i=1; i<=num; i++)

{

printf("%d\n",j);

j=j+2;

}

}

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

#include<stdio.h>

int fact(int);

int main ()

{

int N,f;

printf("enter N number:");

scanf("%d",&N);

f=fact(N);

printf("factorial of %d is %d.",N,f);

return 0;

}

int fact(int num)

{

int factr=1;

while(num)

{

factr=factr\*num;

num--;

}

return factr;

}

1. ***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);

double comb(int,int);

int main()

{

int N,r;

double c;

printf("Enter Nth and r terms:");

scanf("%d%d",&N,&r);

c=comb(N,r);

printf("combination is %.2lf.",c);

return 0;

}

int fact(num)

{

int fctr=1;

while(num)

{

fctr=fctr\*num;

num--;

}

return fctr;

}

double comb(int n,int r)

{

return fact(n)/(fact(r)\*fact(n-r));

}

1. ***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);

double arreng(int,int);

int main()

{

int N,r;

double A;

printf("Enter Nth and r terms:");

scanf("%d%d",&N,&r);

A=arreng(N,r);

printf("arrengement is %.2lf.",A);

return 0;

}

int fact(num)

{

int fctr=1;

while(num)

{

fctr=fctr\*num;

num--;

}

return fctr;

}

double arreng(int n,int r)

{

int a=r>(n-r)?r:(n-r);

return fact(n)/fact(r);

}

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

#include<stdio.h>

int func(int);

int main()

{

int num,ans;

printf("enter a number:");

scanf("%d",&num);

ans=func(num);

printf("%d",ans);

}

int func(int n)

{

int digit,a;

int x=n;

printf("Enter a digit:");

scanf("%d",&digit);

while(n)

{

a=n%10;

if(a==digit)

{

printf("given digit is contains in %d.\n",x);

return digit;

}

n=n/10;

}

if(n==0)

{

printf("given digit is not contains in %d.\n",x);

return digit;

}

}

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).***

#include<stdio.h>

void prime\_fact(int);

int main()

{

int num;

printf("Enter a number:");

scanf("%d",&num);

prime\_fact(num);

return 0;

}

void prime\_fact(int n)

{

int i=2;

while(n!=1)

{

while(n%i==0)

{

n=n/i;

printf("%d ",i);

}

i++;

}

}