**Assignment – 10 A Job Ready Bootcamp in C++, DSA and IOT**

Functions in C Language

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

float areaofcircle(float radius)

{

float area;

area = 3.14\*radius\*radius;

return area;

}

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

float SimpleIntrest(float principal,float rate,float time)

{

float si;

si = (principal\*rate\*time)/100;

return si;

}

**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)**

int checkevenodd(int x)

{

if(x%2==0)

return 1;

else

return 0;

}

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

printNnatural(int x)

{

int i;

printf("The natural numbers are\n");

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

printf("%d ",i);

}

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

printoddNnatural(int x)

{

int i;

printf("The odd natural numbers are\n");

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

printf("%d ",i\*2-1);

}

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

int factofnum(int a)

{

int fact=1;

fact=a;

while(a!=1)

{

a=a-1;

fact\*=a;

}

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)**

int fact(int a)

{

int fact=1;

fact=a;

while(a!=1)

{

a=a-1;

fact\*=a;

}

return fact;

}

int comb(int n,int r)

{

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

}

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

int fact(int a)

{

int fact=1;

fact=a;

while(a!=1)

{

a=a-1;

fact\*=a;

}

return fact;

}

int perm(int n,int r)

{

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

}

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

int checkdigit(int num,int d)

{

int rem=1;

while(num)

{

rem=num%10;

if(rem==d)

return 1;

num = num/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)**

void primefact(int n)

{

int i;

for(i=2;n!=1;i++)

{

while(n%i==0)

{

n=n/i;

printf("%d, ",i);

}

}

}

**11. Write a program to print pascal triangle.**

#include<stdio.h>

int fact(int);

int comb(int,int);

void pascal(int);

int perm(int,int);

int main()

{

pascal(8);

return 0;

}

int fact(int n)

{

int i,fact=1;

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

fact=fact\*i;

return fact;

}

int comb(int n,int r)

{

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

}

int perm(int n,int r)

{

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

}

void pascal(int n)

{

int i,j;

for(i=0;i<=n;i++)

{

for(j=0;j<=i;j++)

{

printf("%d ",comb(i,j));

}

printf("\n");

}

}

