Assignment - 10 (Functions in C Language)

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

double area\_of\_circle(float radius)

{

return 3.14 \* radius \* radius;

}

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

double simpleInterest(float principal, float rate, float time)

{

return principal \* rate \* time / 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)

int isEven(int number)

{

return number % 2 == 0;

}

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

void print\_firstN\_natural\_nums(int N)

{

for(int i = 1; i <= N; i++)

printf("%d ", i);

}

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

void printFirstNOddNaturalNumbers(int N)

{

int i = 1;

while(i <= N)

{

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

i++;

}

}

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

int factorial(int num)

{

int fact = 1;

if(num < 0)

return -1;

else if(num == 0)

return 1;

else

{

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

int factorial(int num)

{

int fact = 1;

if(num < 0)

return -1;

else if(num == 0)

return 1;

else

{

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

fact = fact \* i;

return fact;

}

}

int combination(int n, int r)

{

if(n >= r && r >= 0)

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

else

return -1;

}

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

int factorial(int num)

{

int fact = 1;

if(num < 0)

return -1;

else if(num == 0)

return 1;

else

{

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

fact = fact \* i;

return fact;

}

}

int permutation(int n, int r)

{

if(n >= r && r >= 0)

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

else

return -1;

}

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

int isDigitInNumber(int number, int findDigit)

{

while(number)

{

if(findDigit == number % 10)

return 1;

number = number / 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)

int isPrimeNum(int number)

{

int i;

for(int i = 2; i <= number / 2; i++)

if(number % i == 0)

return 0;

return 1;

}

void printAllPrimeFactorsOfANumber(int number)

{

int x, i;

x = number;

for(i = 2; i <= number / 2; i++)

{

if(isPrimeNum(i))

{

while(x % i == 0)

{

printf("%d, ", i);

x = x / i;

}

if(x == 1)

break;

}

}

printf("\b\b ");

}