MAULANA ABUL KALAM AZAD
UNIVERSITY OF TECHNOLOGY,
WEST BENGAL

Utech

In Pursuit Of Knowledge And Excellence

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL

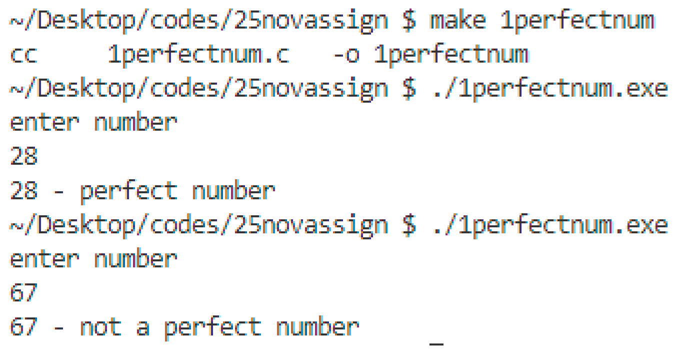
NAME - AMAN ANAND TIWARI
COURSE – BCA (2022 -25)
SEMESTER - I

Practical Assignment for CET I Students

SUBJECT – Programming for Problem Solving
REG. NUMBER - 223001010708
DATE OF ASSIGNMENT – 24 NOV 2022
DATE OF SUBMISSION – 24 NOV 2022

```c
/* 1. Write a program in C to check whether a number is Perfect or
not.*/

#include <stdio.h>

int main(void)
{
   int n,i,sum;
   sum = 0;
   printf("enter number\n");
   scanf("%d",&n);

   for(i=1; i<=n/2; i++)
   {
      if(n%i == 0)
         sum = sum + i;
   }
   if(sum == n)
      printf("%d - perfect number\n",n);
   else
      printf("%d - not a perfect number \n",n);

return 0;
}
```

output -

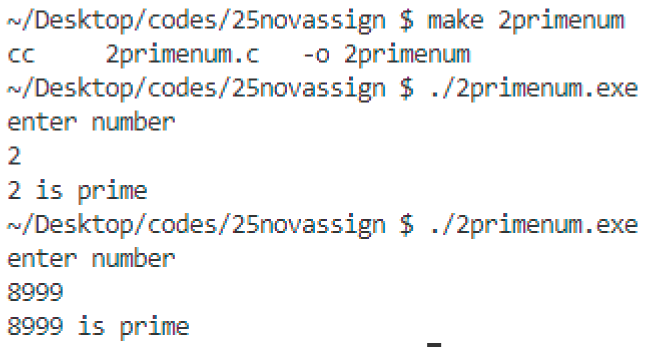


```
~/Desktop/codes/25novassign $ make 1perfectnum
cc     1perfectnum.c    -o 1perfectnum
~/Desktop/codes/25novassign $ ./1perfectnum.exe
enter number
28
28 - perfect number
~/Desktop/codes/25novassign $ ./1perfectnum.exe
enter number
67
67 - not a perfect number
```

```c
/* 2. Write a program in C to check whether a number is
Prime or not.*/

#include <stdio.h>

int main(void)
```

```
{
  int n,i,divisor,t;
  printf("enter number \n");
  scanf("%d",&n);

  divisor = 0;

  for(i=2,t=n/2; i<=t; i++)
  {
    if(n%i == 0)
    {
      divisor += 1;
      break;       /*break as we got atleast one number in
set (1,n) so
            no need to futhur check it is clear that number
is prime*/
    }
  }

  //if total divisors from (2 to n/2) is 0 then prime else
composite
  if(divisor == 0)
    printf("%d is prime \n",n);
  else
    printf("%d is not prime(composite number) \n",n);

  return 0;
}
```

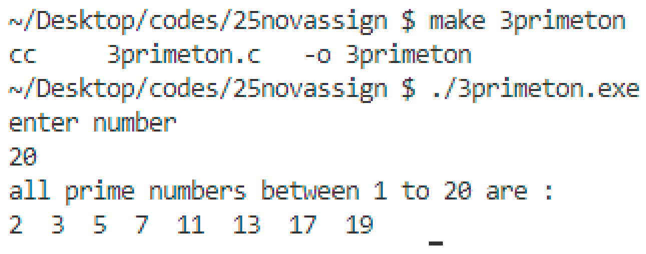
output -



```
~/Desktop/codes/25novassign $ make 2primenum
cc     2primenum.c    -o 2primenum
~/Desktop/codes/25novassign $ ./2primenum.exe
enter number
2
2 is prime
~/Desktop/codes/25novassign $ ./2primenum.exe
enter number
8999
8999 is prime
```

```c
/* 3. Write a program in C to print all prime numbers from
1 to n, n will be taken as user input.*/

#include <stdio.h>

int main(void)
{
  int n,i,j,count;
  count = 0;
  printf("enter number \n");
  scanf("%d",&n);

  printf("all prime numbers between 1 to %d are : \n",n);

  //iterating over numbers from 2 to n as 1 is not prime
  for(i = 2; i<=n; i++)
  {
    count = 0;
     //this loop for checking if a number is prime or not
    for(j=1; j<=i; j++)
    {
      if(i%j == 0)
      {
        count++;
      }
    }
    //printing numbers which are prime
    if(count == 2)
      printf("%d  ",i);
  }
  printf("\n");

  return 0;

}
```

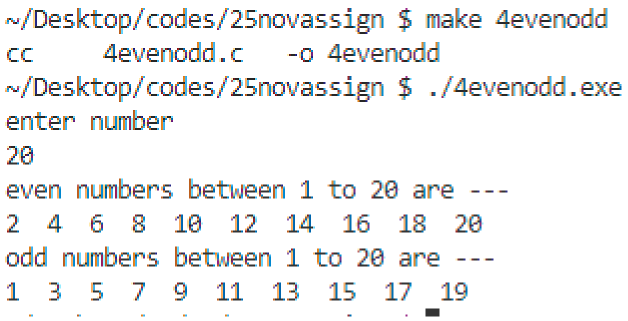
output -



```
~/Desktop/codes/25novassign $ make 3primeton
cc     3primeton.c    -o 3primeton
~/Desktop/codes/25novassign $ ./3primeton.exe
enter number
20
all prime numbers between 1 to 20 are :
2  3  5  7  11  13  17  19
```

```
/* 4. Write a program to print all even and odd numbers
separately from 1 to n,
n will be taken as user input.*/

#include <stdio.h>

int main(void)
{
  int n,i;
  printf("enter number \n");
  scanf("%d",&n);

  printf("even numbers between 1 to %d are --- \n",n);
  for(i = 1; i<=n; i++)
  {
    if(i%2 == 0)
      printf("%d  ",i);
  }
  printf("\n");

  printf("odd numbers between 1 to %d are --- \n",n);
  for(i=1; i<=n; i++)
  {
    if(i%2 != 0)
      printf("%d  ",i);
  }
  printf("\n");

  return 0;
}
```

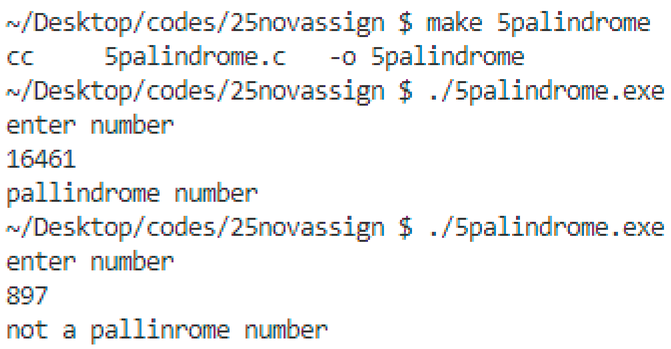
output -



```
~/Desktop/codes/25novassign $ make 4evenodd
cc     4evenodd.c    -o 4evenodd
~/Desktop/codes/25novassign $ ./4evenodd.exe
enter number
20
even numbers between 1 to 20 are ---
2  4  6  8  10  12  14  16  18  20
odd numbers between 1 to 20 are ---
1  3  5  7  9  11  13  15  17  19
```

```c
/* 5. Write a program in C to check whether a number is
Palindrome or not.*/

#include <stdio.h>

int main(void)
{
  int n, rev, temp;
  rev = 0;
  printf("enter number \n");
  scanf("%d",&n);

  temp = n; //this temp will be used for looping so n is not changed

  while(temp!=0)
  {
    rev = 10*rev + temp%10;
    temp = temp/10;
  }

  if(rev==n)
    printf("pallindrome number \n");
  else
    printf("not a pallinrome number \n");


  return 0;
}
```

output -

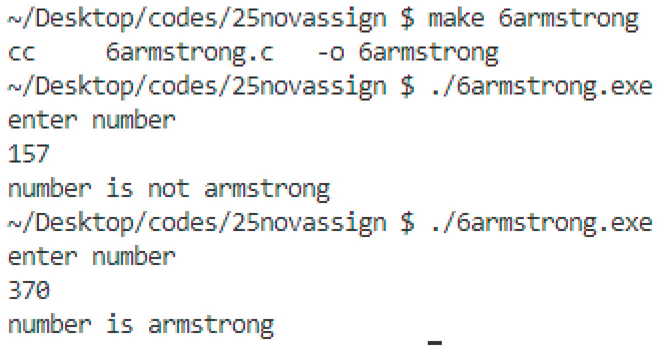


```
~/Desktop/codes/25novassign $ make 5palindrome
cc     5palindrome.c    -o 5palindrome
~/Desktop/codes/25novassign $ ./5palindrome.exe
enter number
16461
pallindrome number
~/Desktop/codes/25novassign $ ./5palindrome.exe
enter number
897
not a pallinrome number
```

```
/* 6. Write a program to check whether a number is
Armstrong or not. */

#include <stdio.h>
#include <math.h>

int main(void)
{
  int n,digit,temp,temp1,sum,r;
  digit =0;
  r =0;
  sum = 0;
  printf("enter number \n");
  scanf("%d",&n);

  temp = n;
  temp1 = n;

  /*counting number of digit for pow() function
  so that we can check armstrong for any number of
digits*/
  while(temp1!=0)
  {
    digit++;
    temp1 /= 10;
  }

  //armstrong logic
  while(temp != 0)
  {
    r = temp%10;
    sum = sum + pow(r,digit);
    temp = temp/10;
  }

  if(sum == n)   //as n is unchanged and temp was changed
    printf("number is armstrong \n");
  else
    printf("number is not armstrong \n");

  return 0;
```

```c
}
```

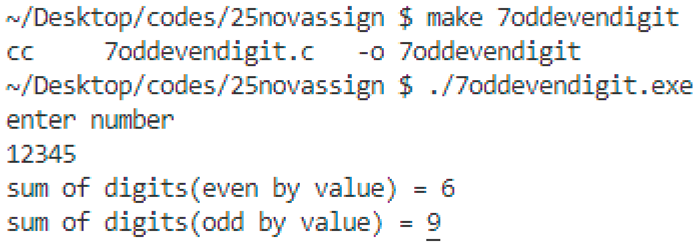
output -



```
~/Desktop/codes/25novassign $ make 6armstrong
cc     6armstrong.c   -o 6armstrong
~/Desktop/codes/25novassign $ ./6armstrong.exe
enter number
157
number is not armstrong
~/Desktop/codes/25novassign $ ./6armstrong.exe
enter number
370
number is armstrong
```

```c
/* 7. Write a program to find the sum of even and odd
digits of a number separately.*/

#include <stdio.h>

int main(void)
{
  int n,odd_sum,even_sum,r;
  odd_sum = 0;
  even_sum = 0;
  printf("enter number \n");
  scanf("%d",&n);
  while(n!=0)
  {
    r = n%10;
    if(r%2 == 0)
      even_sum = even_sum+r;
    else
      odd_sum = odd_sum + r;

    n = n/10;
  }
  printf("sum of digits(even by value) = %d \n",even_sum);
  printf("sum of digits(odd by value) = %d \n",odd_sum);
return 0;
}
```

output -

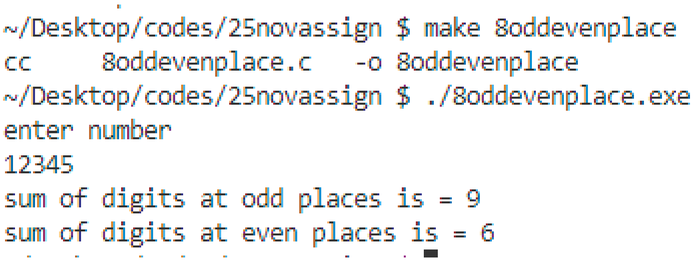


```
~/Desktop/codes/25novassign $ make 7oddevendigit
cc     7oddevendigit.c   -o 7oddevendigit
~/Desktop/codes/25novassign $ ./7oddevendigit.exe
enter number
12345
sum of digits(even by value) = 6
sum of digits(odd by value) = 9
```

```
/* 8. Write a program to find the sum of even and odd
place digit of a number separately*/



#include <stdio.h>

int main(void)
{
  int n,oddsum,evensum,count,rev;
  oddsum=0;
  evensum=0;
  count =1;
  printf("enter number \n");
  scanf("%d",&n);

  //reversing number
  rev =0;
  while(n!=0)
  {
    rev = rev*10 + n%10;
    n = n/10;
  }

  while(rev != 0)
  {
    if(count%2 != 0)
      oddsum += rev%10;
    else
      evensum += rev%10;

    count++;
    rev = rev/10;
  }



  printf("sum of digits at odd places is = %d \n",oddsum);
  printf("sum of digits at even places is= %d \
n",evensum);
```

```
  return 0;
}
```

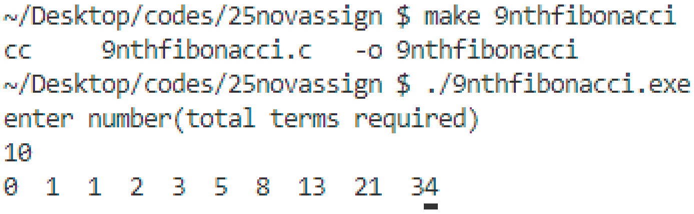
output -



```
~/Desktop/codes/25novassign $ make 8oddevenplace
cc     8oddevenplace.c    -o 8oddevenplace
~/Desktop/codes/25novassign $ ./8oddevenplace.exe
enter number
12345
sum of digits at odd places is = 9
sum of digits at even places is = 6
```

```
/* 9. Write a program to print Fibonacci series upto nth
Term.*/

#include <stdio.h>

int main(void)
{
  int n,sum,a,b,i;
  a=0, b=1;
  printf("enter number(total terms required) \n");
  scanf("%d",&n);

  //fibonacci printing
  printf("%d  %d  ",a,b);

  for(i=2; i<n; i++)
  {
    sum = a+b;
    printf("%d  ",sum);
    a = b;
    b = sum;
  }
  printf("\n");

  return 0;
}
```

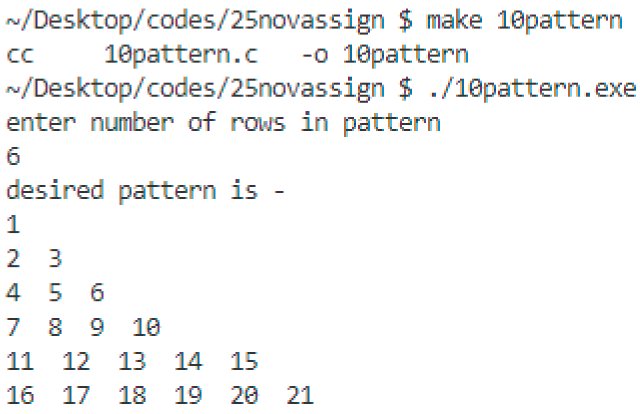
output -



```
~/Desktop/codes/25novassign $ make 9nthfibonacci
cc     9nthfibonacci.c    -o 9nthfibonacci
~/Desktop/codes/25novassign $ ./9nthfibonacci.exe
enter number(total terms required)
10
0  1  1  2  3  5  8  13  21  34
```

```
/* 10. Write a program to print the following pattern:
            1
            2      3
            4      5      6
            7      8      9       10         */


#include <stdio.h>

int main(void)
{
  int n,i,j,c;
  printf("enter number of rows in pattern \n");
  scanf("%d",&n);

  c = 0;

  printf("desired pattern is - \n");


  //loop for each row
  for(i=0; i<n; i++)
  {

    //loop for each column in the current row
    for(j=0; j<=i; j++)
    {
      c = c+1;
      printf("%d  ",c);
    }

    printf("\n");
  }
  return 0;
}
```

output -



```
~/Desktop/codes/25novassign $ make 10pattern
cc     10pattern.c   -o 10pattern
~/Desktop/codes/25novassign $ ./10pattern.exe
enter number of rows in pattern
6
desired pattern is -
1
2  3
4  5  6
7  8  9  10
11  12  13  14  15
16  17  18  19  20  21
```