UNIVERSITY OF JAMMU

BHADERWAH CAMPUS

BACHELORS IN COMPUTER APPLICATION

(BCA-1ST SEMESTER)

PRACTICAL FILE

PROBLEM SOLVING USING C

SUBMITTED BY: SUBMITTED TO: Mohd
Sami Mr. Abhay Khajuria

ACKNOWLEDGEMENT

I would like to express my gratitude to all those people who helped me to complete this practical file. I want to thank Department of computer science and IT for providing me all the tools and resources for the completion of my practical file.

I have further more to thank our respected teacher Mr. Abhay Khajuria who encouraged me to move ahead and complete my work in time. I am also bound to our Assistant Programmer Mr. Parvesh Singh whose help, simulating suggestions and encouragement helped me all the time during my work.

At last, I want to thank my classmates and friends for their help, support, interest and valuable hints. I want to thank once again all the above mentioned persons for their time and help.

Thank you.

Mohd Sami

BCA 1st Semester

113040022

CERTIFICATE

This is to certify that MOHD SAMI student of BCA 1st semester bearing university Roll no. 113040022 has completed the required number of practicals of Problem solving using C under the guidance of class teacher MR. Abhay Khajuria from the Department of Computer science and IT, Bhaderwah campus, University of Jammu.

Signature

(Subject I/C)

INDEX

27 WAP to C program to create prime number by creating a function.

28 WAP to display prime numbers between two intervals using functions and for loop.

29 WAP to check whether numbers can be expressed as the sum of two prime numbers.

30 WAP to find sum of natural numbers using recursion.

31 WAP to calculate factorial of a number using recursion.

32 WAP to find HCF of two numbers.

33 WAP to reverse a stack using recursion, without using any loop.

34 WAP to calculate power of a number.

35 WAP to find largest number in an array.

36 WAP to check maximum and minimum size of an array.

37 WAP to calculate average of all elements of an array.

38 WAP to find common array elements between two arrays.

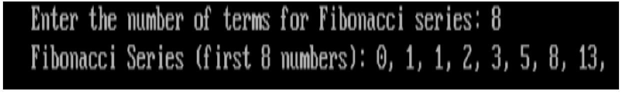
39 WAP to copy all the elements of an array to another.

40 WAP to find determinant of 2x2 Matrix.

PROGRAM 1:- WAP to print Fibonacci series.

```
#include <stdio.h>

#include <conio.h>
int main()
{
  int n, first = 0, second = 1, next, i;
clrscr();

printf("Enter the number of terms for Fibonacci series: ");
scanf("%d", &n);

printf("Fibonacci Series (first %d numbers): ", n);

   for (i = 0; i< n; i++)
   {
     printf("%d, ", first);

     next = first + second;
     first = second;
     second = next;
   }
printf("\n");
getch();
   return 0;
}
```

Output:-



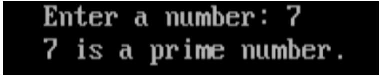
```
Enter the number of terms for Fibonacci series: 8
Fibonacci Series (first 8 numbers): 0, 1, 1, 2, 3, 5, 8, 13,
```

***

Program 2:- WAP to check prime number.

```
#include <stdio.h>
#include <conio.h>
int main()
{
   int num, isPrime = 1, i;
clrscr();

printf("Enter a number: ");
scanf("%d", &num);
if (num<= 1)
   {
     isPrime = 0;
   }
   else
   {
     for (i = 2; i * i<= num&&isPrime; i++)
   {
        if (num % i == 0)
        {
          isPrime = 0003B
        }
   }
   }

   if (isPrime)
   {
     printf("%d is a prime number.\n", num);
   }
   else
   {
     printf("%d is not a prime number.\n", num);
   }
getch();
   return 0;
}
```

Output:-



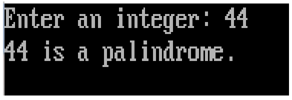
```
 Enter a number: 7
 7 is a prime number.
```

***

Program 3:- WAP to check palindrome number.

```
#include <stdio.h>
#include <conio.h>

int main()
{
  int num, reversedNum = 0, originalNum;
clrscr();

printf("Enter an integer: ");
scanf("%d", &num);

originalNum = num;

  while (num)
  {
    reversedNum = reversedNum * 10 + num % 10;
    num /= 10;
  }

  if (originalNum == reversedNum)
    printf("%d is a palindrome.\n", originalNum);
  else
    printf("%d is not a palindrome.\n", originalNum);

getch();
  return 0;
}
```

Output:-

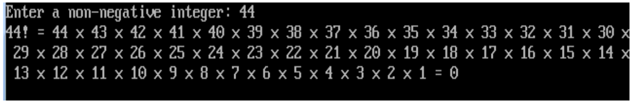


```
Enter an integer: 44
44 is a palindrome.
```

***

Program 4:- WAP to print factorial of a number.

```
#include <stdio.h>
#include <conio.h>

int main()
{
   int num, factorial = 1, i;
clrscr();

printf("Enter a non-negative integer: ");
scanf("%d", &num);

   if (num< 0)
   {
     printf("Factorial is not defined for negative numbers.\n");
   }
   else
   {
     printf("%d! = ", num);
     for (i = num; i>= 1; –i)
     {
        factorial *= i;

     printf("%d", i);
        if (i> 1)
        {
          printf(" x ");
        }
     }

     printf(" = %d\n", factorial);
   }
getch();
   return 0;
}
```



```
Enter a non-negative integer: 44
44! = 44 x 43 x 42 x 41 x 40 x 39 x 38 x 37 x 36 x 35 x 34 x 33 x 32 x 31 x 30 x
 29 x 28 x 27 x 26 x 25 x 24 x 23 x 22 x 21 x 20 x 19 x 18 x 17 x 16 x 15 x 14 x
 13 x 12 x 11 x 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 0
```

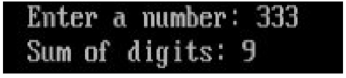
Output:-

Program 5:- WAP to print sum of digits.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
{
    int num, digit, sum = 0;

clrscr();

printf("Enter a number: ");
scanf("%d", &num);

    while (num> 0)
{
       digit = num % 10;
       sum += digit;
       num /= 10;
    }

printf("Sum of digits: %d", sum);

getch();

    return 0;
}
```

Output:-



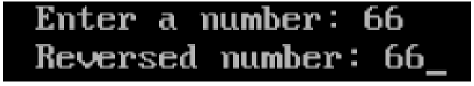
```
Enter a number: 333
Sum of digits: 9
```

***

Program 6:- WAP to reverse given number.

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int num, reversedNum = 0, remainder;

clrscr();
printf("Enter a number: ");
scanf("%d", &num);

   while (num != 0)
{
      remainder = num % 10;
reversedNum = reversedNum * 10 + remainder;
num /= 10;
   }

printf("Reversed number: %d", reversedNum);

getch();

   return 0;
}
```

Output:-



```
Enter a number: 66
Reversed number: 66_
```

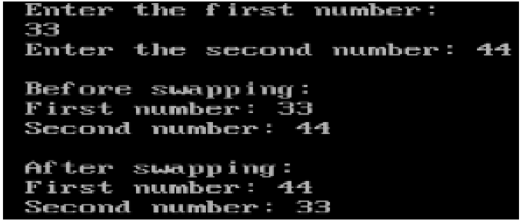
***

Program 7:- WAP to swap two numbers without using third variable.

```
#include <stdio.h>
#include <conio.h>

int main() {
   int num1, num2;

clrscr();

printf("Enter the first number: ");
scanf("%d", &num1);

printf("Enter the second number: ");
scanf("%d", &num2);

printf("\nBefore swapping:\n");
printf("First number: %d\n", num1);
printf("Second number: %d\n", num2);

   num1 = num1 + num2;
   num2 = num1 - num2;
```

```
    num1 = num1 - num2;

printf("\nAfter swapping:\n");
printf("First number: %d\n", num1);
printf("Second number: %d\n", num2);

getch();
    return 0;
}
```

Output:-



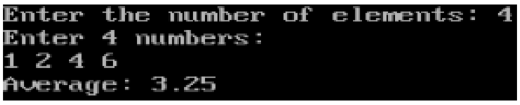
```
Enter the first number:
33
Enter the second number: 44

Before swapping:
First number: 33
Second number: 44

After swapping:
First number: 44
Second number: 33
```

***

Program 8:- WAP to find average of n numbers.

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int n, i;
    float sum = 0, average;

clrscr();

printf("Enter the number of elements: ");
scanf("%d", &n);

printf("Enter %d numbers:\n", n);
    for (i = 0; i< n; i++)
{
        float num;
scanf("%f", &num);
        sum += num;
    }

    average = sum / n;

printf("Average: %.2f", average);
getch();

   return 0;
}
```

Output:-



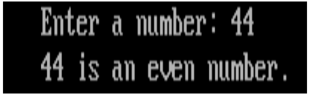
```
Enter the number of elements: 4
Enter 4 numbers:
1 2 4 6
Average: 3.25
```

***

Program 9:- WAP to checking for odd and even numbers using bitwise operators.

```
#include <stdio.h>
#include <conio.h>

int main() {
   int num;

clrscr();

printf("Enter a number: ");
scanf("%d", &num);

   if (num& 1) {
printf("%d is an odd number.", num);
   } else {
printf("%d is an even number.", num);
   }

getch();

   return 0;
}
```

Output:-



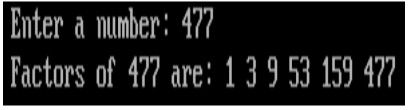
```
Enter a number: 44
44 is an even number.
```

***

Program 10:- WAP to find factors of a number. +

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int num, i;

clrscr();

printf("Enter a number: ");
scanf("%d", &num);

printf("Factors of %d are: ", num);

    for (i = 1; i<= num; i++)
{
        if (num % i == 0)
{
printf("%d ", i);
        }
    }

getch();

    return 0;
}
```

Output:-



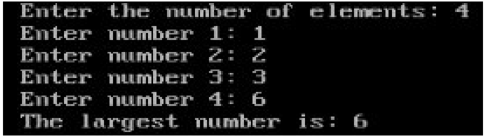
```
Enter a number: 477
Factors of 477 are: 1 3 9 53 159 477
```

***

Program 11:- WAP to find the largest number among n input numbers.

```
#include <stdio.h>
#include <conio.h>

int main()
{
   int n, i;
   int largest, currentNumber;
clrscr();

printf("Enter the number of elements: ");
scanf("%d", &n);

printf("Enter number 1: ");
scanf("%d", &largest);

   for (i = 2; i<= n; i++)
   {
printf("Enter number %d: ", i);
scanf("%d", &currentNumber);

      if (currentNumber> largest)
      {
         largest = currentNumber;
      }
   }
printf("The largest number is: %d", largest);
getch();
return 0;
}
```

Output:-



```
Enter the number of elements: 4
Enter number 1: 1
Enter number 2: 2
Enter number 3: 3
Enter number 4: 6
The largest number is: 6
```

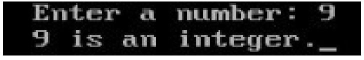
***

Program 12:- WAP to check if input number is int or float.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
{
   char input[50];

clrscr();

printf("Enter a number: ");
scanf("%s", input);

   if (strchr(input, '.') != NULL)
   {
printf("%s is a floating-point number.", input);
   } else
   {
printf("%s is an integer.", input);
   }

getch();

   return 0;
}
```

Output:-



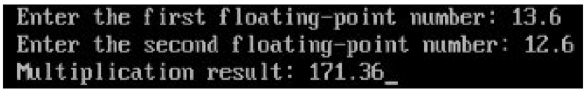
```
Enter a number: 9
9 is an integer._
```

***

Program 13:- WAP to Multiply two floating-point numbers.

```
#include <stdio.h>
#include <conio.h>

int main()
{
   float num1, num2, result;

clrscr();

printf("Enter the first floating-point number: ");
scanf("%f", &num1);
printf("Enter the second floating-point number: ");
scanf("%f", &num2);

   result = num1 * num2;

printf("Multiplication result: %.2f", result);

getch();

   return 0;
}
```

Output:-



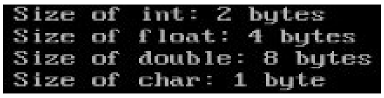
```
Enter the first floating-point number: 13.6
Enter the second floating-point number: 12.6
Multiplication result: 171.36_
```

***

Program 14:- WAP to find the size of int, float, double and char.

```
#include <stdio.h>
#include <conio.h>

int main() {
clrscr();

printf("Size of int: %d bytes\n", sizeof(int));
printf("Size of float: %d bytes\n", sizeof(float));
printf("Size of double: %d bytes\n", sizeof(double));
printf("Size of char: %d byte\n", sizeof(char));

getch();

   return 0;
}
```

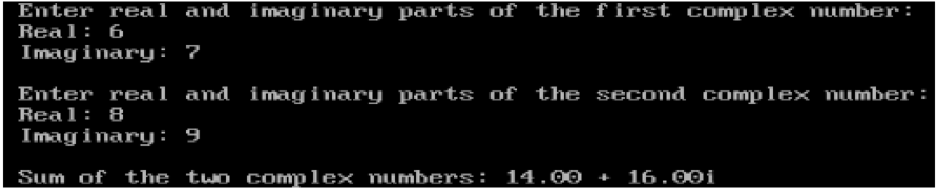
Output:-



```
Size of int: 2 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte
```

```
}
```

Output:-

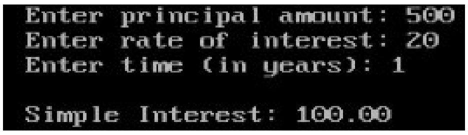


```
Enter real and imaginary parts of the first complex number:
Real: 6
Imaginary: 7

Enter real and imaginary parts of the second complex number:
Real: 8
Imaginary: 9

Sum of the two complex numbers: 14.00 + 16.00i
```

Program 16:- WAP to find simple interest.

```
#include <stdio.h>
#include <conio.h>

int main()
{
   float principal, rate, time, simpleInterest;

clrscr();

printf("Enter principal amount: ");
scanf("%f", &principal);

printf("Enter rate of interest: ");
scanf("%f", &rate);

printf("Enter time (in years): ");
scanf("%f", &time);

simpleInterest = (principal * rate * time) / 100;

printf("\nSimple Interest: %.2f\n", simpleInterest);

getch();

   return 0;
}
```

Output:-



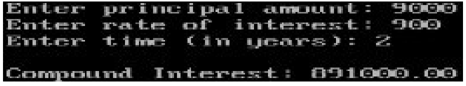
```
Enter principal amount: 500
Enter rate of interest: 20
Enter time (in years): 1

Simple Interest: 100.00
```

***

Program 17:- WAP to find compound interest.

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

int main()
{
   float principal, rate, time, compoundInterest;

clrscr();

printf("Enter principal amount: ");
scanf("%f", &principal);

printf("Enter rate of interest: ");
scanf("%f", &rate);

printf("Enter time (in years): ");
scanf("%f", &time);

compoundInterest = principal * pow(1 + rate / 100, time) - principal;

printf("\nCompound Interest: %.2f\n", compoundInterest);

getch();

   return 0;
}
```

Output:-



```
Enter principal amount: 9000
Enter rate of interest: 900
Enter time (in years): 2

Compound Interest: 891000.00
```

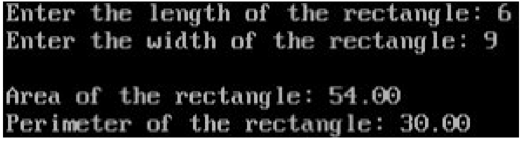
***

Program 18:- WAP to find area and perimeter of rectangle.

```
#include <stdio.h>
```

```
#include <conio.h>

int main() {
clrscr();

   float length, width, area, perimeter;

printf("Enter the length of the rectangle: ");
scanf("%f", &length);

printf("Enter the width of the rectangle: ");
scanf("%f", &width);

   area = length * width;
   perimeter = 2 * (length + width);

printf("\nArea of the rectangle: %.2f\n", area);
printf("Perimeter of the rectangle: %.2f\n", perimeter);

getch();

   return 0;
}
```

Output:-



```
Enter the length of the rectangle: 6
Enter the width of the rectangle: 9

Area of the rectangle: 54.00
Perimeter of the rectangle: 30.00
```

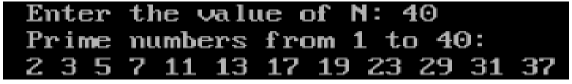
***

Program 19:-WAP to print prime numbers from 1 toN.

```
#include <stdio.h>
int isPrime(int num);
int main() {
   int N, i;
clrscr();
printf("Enter the value of N: ");
scanf("%d", &N);
printf("Prime numbers from 1 to %d:\n", N);
   for (i = 2; i<= N; i++) {
      if (isPrime(i)) {
```

```
     printf("%d ", i);
     }
   }
getch();
   return 0;
}
int isPrime(int num) {
   int i;
   if (num<= 1) {
     return 0;
   }
   for (i = 2; i * i<= num; i++) {
     if (num % i == 0) {
        return 0;
     }
   }

   return 1;
}
```

Output:-



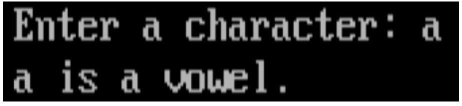
```
 Enter the value of N: 40
 Prime numbers from 1 to 40:
 2 3 5 7 11 13 17 19 23 29 31 37
```

***

Program 20:- WAP to find characters is vowel or not.

```
#include <stdio.h>
#include <conio.h>

int main()
{
   char ch;

clrscr();

printf("Enter a character: ");
scanf(" %c", &ch);

   if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch== 'u' ||
ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
{
printf("%c is a vowel.\n", ch);
   }
else
{
printf("%c is not a vowel.\n", ch);
   }

getch();

   return 0;
}
```

Output:-



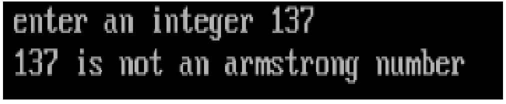
```
Enter a character: a
a is a vowel.
```

***

Program 21:-WAP to check whether the number is Armstrong or not

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int a,a1,rmd,i=0,r=0;
clrscr();
printf("enter an integer:");
scanf("%d",&a);
a1=a;
for(a1=a;a1!=0;i++)
{
a1/=10;
}
for(a1=a;a1!=0;a1/=10)
{
rmd=a1%10;
r+= pow(rmd,i);
}
if((int)r==a)
  printf("%d is an armstrong number",a);
else
  printf("%d is not an armstrong number",a);
getch();
}
```

Output:-

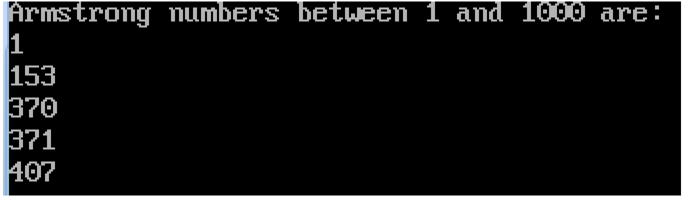


```
enter an integer 137
137 is not an armstrong number
```

Program 22:- WAP to display Armstrong number between 1-1000.

```
#include <stdio.h>
#include <conio.h>

int main()
{
   int num, originalNum, sum = 0, digit;
   clrscr();

   printf("Enter an integer: ");
   scanf("%d", &num);

   originalNum = num;

   while (num > 0)
   {
      digit = num % 10;
      sum += digit * digit * digit;
      num /= 10;
   }

   if (sum == originalNum)
      printf("%d is an Armstrong number.\n", originalNum);
   else
      printf("%d is not an Armstrong number.\n", originalNum);
   getch();
   return 0;
}
```

Output:-

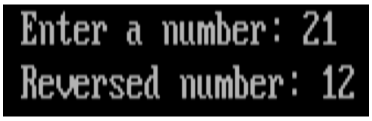


```
Armstrong numbers between 1 and 1000 are:
1
153
370
371
407
```

Program 23:- WAP to display the reverse number.

```
#include <stdio.h>
#include <conio.h>
int main()
{
   int num, reversed = 0;
   clrscr();
   printf("Enter a number: ");
   scanf("%d", &num);

   while (num != 0)
   {
      reversed = reversed * 10 + num % 10;
      num /= 10;
   }
   printf("Reversed number: %d", reversed);
   getch();
   return 0;
}
```

Output:-

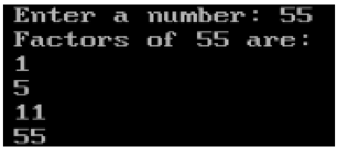


```
Enter a number: 21
Reversed number: 12
```

Program 24:- WAP to display all factors of a number.

```
#include <stdio.h>
#include <conio.h>
int main()
{
   int num, i;
   clrscr();
   printf("Enter a number: ");
   scanf("%d", &num);

   printf("Factors of %d are:\n", num);
   for (i = 1; i <= num; ++i)
   {
     if (num % i == 0)
     {
        printf("%d\n", i);
     }
   }
   getch();
   return 0;
}
```

Output:-



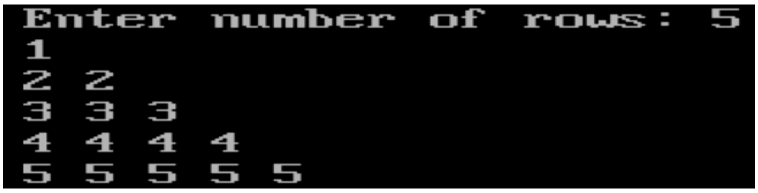
```
Enter a number: 55
Factors of 55 are:
1
5
11
55
```

Program 25:- WAP to display half pyramid

```
#include <stdio.h>
#include <conio.h>
int main()
{
   int rows, i, j;
```

```
    clrscr();
    printf("Enter number of rows: ");
    scanf("%d", &rows);

    for (i = 1; i <= rows; ++i)
    {
      for (j = 1; j <= i; ++j)
      {
          printf("%d ", i);
      }
      printf("\n");
    }
    getch();
    return 0;
}
```

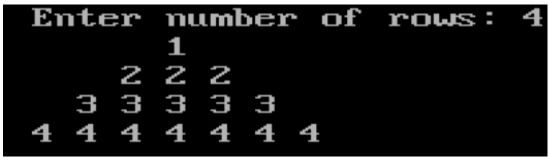
Output:-



```
Enter number of rows: 5
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Program 26:- WAP to display full pyramid.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int rows, i, k, space, num = 1;
    clrscr();
    printf("Enter number of rows: ");
    scanf("%d", &rows);
    for (i = 1, k = 0; i <= rows; ++i, k = 0)
    {
        for (space = 1; space <= rows - i; ++space)
        {
           printf("  ");
        }
        while (k != 2 * i - 1)
        {
           printf("%d ", num);
           ++k;
        }
        num++;
        printf("\n");
     }    getch();    return 0; }
```



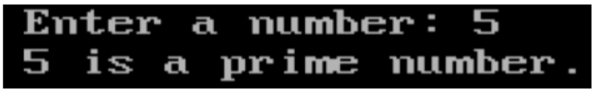
```
Enter number of rows: 4
          1
       2 2 2
    3 3 3 3 3
 4 4 4 4 4 4 4
```

Output:-

Program 27:- WAP to C program to create prime number by creating a function .

```
#include <stdio.h>
#include <conio.h>

int isPrime(int num);

int main()
{
   int number;

   clrscr();
   printf("Enter a number: ");
   scanf("%d", &number);

   if (isPrime(number))
   {
      printf("%d is a prime number.", number);
```

```
    } else
    {
        printf("%d is not a prime number.", number);
    }

    getch();
    return 0;
}

int isPrime(int num)
{
    int i;

    if (num < 2)
    {
        return 0;
    }

    for (i = 2; i <= num / 2; ++i)
    {
        if (num % i == 0)
        {
            return 0;
        }
    }

    return 1;
}
```

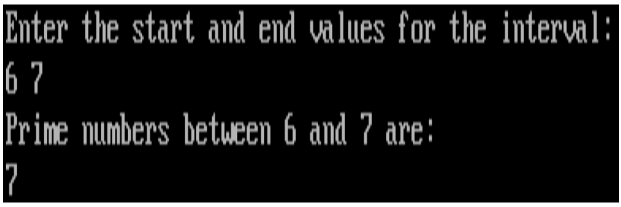


```
Enter a number: 5
5 is a prime number.
```

Output:-

Program 28:- WAP to display prime numbers between two intervals using functions and for loop.

```
#include <stdio.h>
#include <conio.h>

int isPrime(int num);

int main()
{
    int start, end, i;

    clrscr();
    printf("Enter the start and end values for the interval: ");
    scanf("%d %d", &start, &end);

    printf("Prime numbers between %d and %d are:\n", start, end);
    for (i = start; i <= end; ++i)
    {
        if (isPrime(i))
        {
            printf("%d\n", i);
        }
    }

    getch();
    return 0;
}                               c

int isPrime(int num)
{
    int i, isPrimeFlag = 1;

    if (num < 2)
    {
        isPrimeFlag = 0;
    }

    for (i = 2; i <= num / 2; ++i)
    {
        if (num % i == 0)
        {
            isPrimeFlag = 0;
            break;
        }
    }

    return isPrimeFlag;
}
```



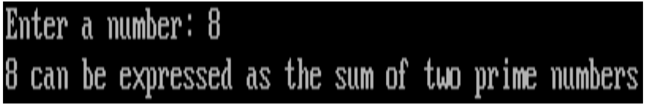
```
Enter the start and end values for the interval:
6 7
Prime numbers between 6 and 7 are:
7
```

Output:-

Program 29:- WAP to check whether numbers can be expressed as the sum of two prime numbers.

```
#include <stdio.h>
#include <conio.h>

int isPrime(int num);

int main()
{
    int number, i;
    int flag = 0;

    clrscr();
    printf("Enter a number: ");
    scanf("%d", &number);

    for (i = 2; i <= number / 2; ++i)
    {
      if (isPrime(i)) {
         if (isPrime(number - i))
         {
           flag = 1;
           break;
         }
      }
    }

    if (flag)
    {
      printf("%d can be expressed as the sum of two prime numbers.", number);
    }
    else
    {
      printf("%d cannot be expressed as the sum of two prime numbers.", number);
    }

    getch();
    return 0;
}

int isPrime(int num) {
    int i, isPrimeFlag = 1;

    if (num < 2) {
      isPrimeFlag = 0;
    }

    for (i = 2; i <= num / 2; ++i) {
        if (num % i == 0) {
            isPrimeFlag = 0;
            break;
        }
    }

    return isPrimeFlag;
}
```

Output:-

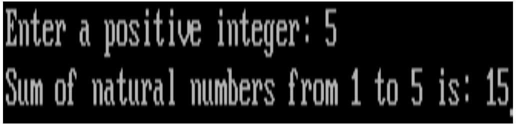


```
Enter a number: 8
8 can be expressed as the sum of two prime numbers
```

Program 30:- WAP to find sum of natural numbers using recursion.

```
#include <stdio.h>
#include <conio.h>

int sumOfNaturalNumbers(int n);

int main()
{
   int num;

   clrscr();
   printf("Enter a positive integer: ");
   scanf("%d", &num);

   printf("Sum of natural numbers from 1 to %d is: %d", num,
sumOfNaturalNumbers(num));

   getch();
   return 0;
}

int sumOfNaturalNumbers(int n)
{
   if (n == 0)
   {
     return 0;
   }
   else
   {
     return n + sumOfNaturalNumbers(n - 1);
   }
}
```

Output:-

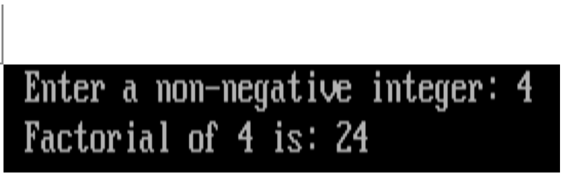


```
Enter a positive integer: 5
Sum of natural numbers from 1 to 5 is: 15
```

Program 31:- WAP to calculate factorial of a number using recursion.

```
#include <stdio.h>
#include <conio.h>

int factorial(int n);

int main()
{
   int num;

   clrscr();
   printf("Enter a non-negative integer: ");
   scanf("%d", &num);

   printf("Factorial of %d is: %d", num, factorial(num));

   getch();
   return 0;
}
int factorial(int n)
{
   if (n == 0 || n == 1)
   {
     return 1;
   }
   else
   {
     return n * factorial(n - 1);
   }
}
```

Outout

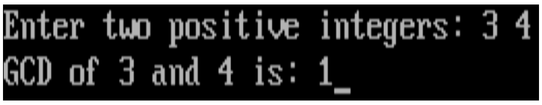


```
Enter a non-negative integer: 4
Factorial of 4 is: 24
```

Program 32:- WAP to find HCF of two numbers.

```
#include <stdio.h>
#include <conio.h>

int findGCD(int num1, int num2);

int main()
{
   int num1, num2;

   clrscr();
   printf("Enter two positive integers: ");
   scanf("%d %d", &num1, &num2);

   printf("GCD of %d and %d is: %d", num1, num2, findGCD(num1, num2));

   getch();
   return 0;
}

int findGCD(int num1, int num2)
{
   while (num1 != num2)
   {
     if (num1 > num2)
        num1 -= num2;
     else
        num2 -= num1;
   }
   return num1;
}
```

Output

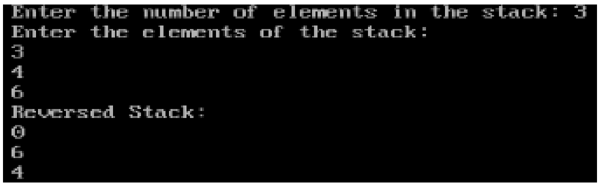


```
Enter two positive integers: 3 4
GCD of 3 and 4 is: 1_
```

Program 33:- WAP to reverse a stack using recursion, without using any loop.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define MAX_SIZE 100
struct Stack
{
   int items[MAX_SIZE];
   int top;
};
void initializeStack(struct Stack *s);
int isEmpty(struct Stack *s);
void push(struct Stack *s, int value);
int pop(struct Stack *s);
void reverseStack(struct Stack *s, int value);
int main()
{
   int n,i;
   struct Stack stack;
   clrscr();
   initializeStack(&stack);
   printf("Enter the number of elements in the stack: ");
   scanf("%d", &n);
   printf("Enter the elements of the stack:\n");
   for (i = 0; i < n; ++i)
   {
     int element;
     scanf("%d", &element);
     push(&stack, element);
   }
   reverseStack(&stack, 0);
```

```
    printf("Reversed Stack:\n");
    while (!isEmpty(&stack))
    {
        printf("%d\n", pop(&stack));
    }
    getch();
    return 0;
}
void initializeStack(struct Stack *s)
{
    s->top = -1;
}
int isEmpty(struct Stack *s)
{
    return s->top == -1;
}
void push(struct Stack *s, int value)
{
    if (s->top == MAX_SIZE - 1)
    {
        printf("Stack overflow\n");
        return;
    }
    s->items[++s->top] = value;
}

int pop(struct Stack *s)
{
    if (isEmpty(s))
    {
        printf("Stack underflow\n");
        return -1;
    }
    return s->items[s->top–];
}
void reverseStack(struct Stack *s, int value)
{
    if (!isEmpty(s))
    {
        int temp = pop(s);
        reverseStack(s, temp);
        push(s, value);
```



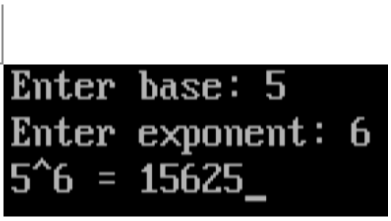
```
Enter the number of elements in the stack: 3
Enter the elements of the stack:
3
4
6
Reversed Stack:
0
6
4
```

```
    }
} output
```

Program 34:- WAP to calculate power of a number.

```
#include <stdio.h>
#include <conio.h>

int power(int base, int exponent);

int main()
{
   int base, exponent;

   clrscr();
   printf("Enter base: ");
   scanf("%d", &base);

   printf("Enter exponent: ");
   scanf("%d", &exponent);

   printf("%d^%d = %d", base, exponent, power(base, exponent));

   getch();
   return 0;
}

int power(int base, int exponent) {
   if (exponent == 0) {
      return 1;
   } else {
      return base * power(base, exponent - 1);
   }
}
```

Output



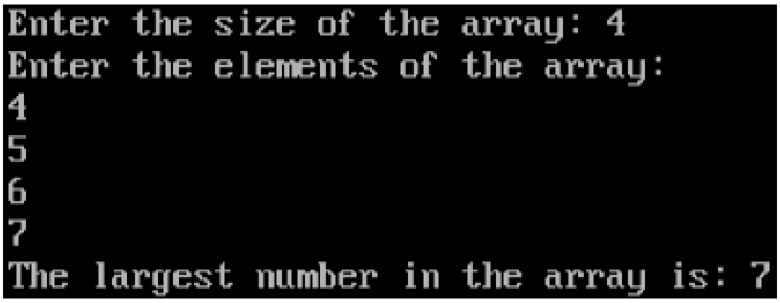
```
Enter base: 5
Enter exponent: 6
5^6 = 15625_
```

Program 35:- WAP to find largest number in an array.

```
#include <stdio.h>

#include <conio.h>


#define MAX_SIZE 100


int findLargest(int arr[], int size);


int main()

{

   int arr[MAX_SIZE];

   int size, i;


   clrscr();

   printf("Enter the size of the array: ");

   scanf("%d", &size);


   printf("Enter the elements of the array:\n");

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

   {

     scanf("%d", &arr[i]);

   }


   printf("The largest number in the array is: %d", findLargest(arr, size));


   getch();

   return 0;

}


int findLargest(int arr[], int size)

{

   int max = arr[0], i;
```

```
    for (i = 1; i < size; ++i)
    {
        if (arr[i] > max)
        {
            max = arr[i];
        }
    }

    return max;
}
```

Output

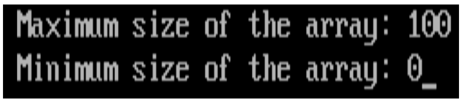


```
Enter the size of the array: 4
Enter the elements of the array:
4
5
6
7
The largest number in the array is: 7
```

Program 36:- WAP to check maximum and minimum size of an array.

```
#include <stdio.h>
#include <conio.h>

#define MAX_SIZE 100

int main()
{
   int arr[MAX_SIZE];

   clrscr();

   printf("Maximum size of the array: %d\n", MAX_SIZE);
   printf("Minimum size of the array: 0");

   getch();
   return 0;
}
```

Output



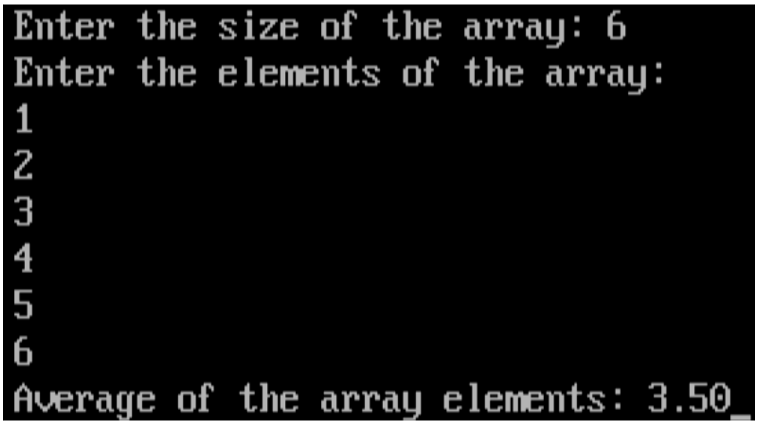
```
Maximum size of the array: 100
Minimum size of the array: 0_
```

Program 37:- WAP to calculate average of all elements of an array.

```
#include <stdio.h>

#include <conio.h>


#define MAX_SIZE 100


float calculateAverage(int arr[], int size);


int main()

{

   int arr[MAX_SIZE];

   int size, i;


   clrscr();

   printf("Enter the size of the array: ");

   scanf("%d", &size);


   printf("Enter the elements of the array:\n");

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

   {

     scanf("%d", &arr[i]);

   }


   printf("Average of the array elements: %.2f", calculateAverage(arr, size));


   getch();

   return 0;

}


float calculateAverage(int arr[], int size)

{

   int sum = 0, i;
```

```
    for (i = 0; i < size; ++i)
    {
        sum += arr[i];
    }

    return (float)sum / size;
}
```

Output



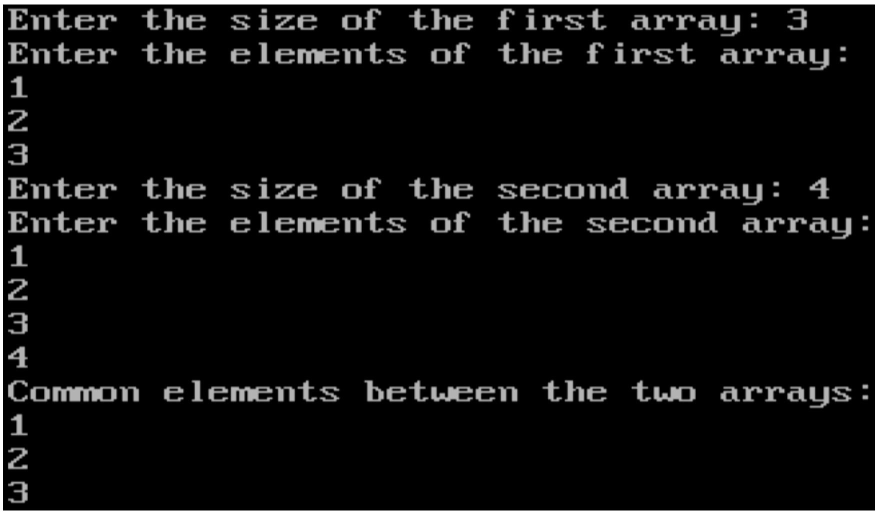
```
Enter the size of the array: 6
Enter the elements of the array:
1
2
3
4
5
6
Average of the array elements: 3.50_
```

Program 38:- WAP to find common array elements between two arrays.

```
#include <stdio.h>

#include <conio.h>


#define MAX_SIZE 100


void findCommonElements(int arr1[], int size1, int arr2[], int size2);


int main()

{

   int arr1[MAX_SIZE], arr2[MAX_SIZE];

   int size1, size2, i;


   clrscr();

   printf("Enter the size of the first array: ");

   scanf("%d", &size1);


   printf("Enter the elements of the first array:\n");

   for (i = 0; i < size1; ++i)

   {

     scanf("%d", &arr1[i]);

   }


   printf("Enter the size of the second array: ");

   scanf("%d", &size2);


   printf("Enter the elements of the second array:\n");


   for (i = 0; i < size2; ++i) {

     scanf("%d", &arr2[i]);

   }
```

```
    findCommonElements(arr1, size1, arr2, size2);

    getch();
    return 0;
}


void findCommonElements(int arr1[], int size1, int arr2[], int size2)
{
    int i, j;
    printf("Common elements between the two arrays:\n");

    for (i = 0; i < size1; ++i) {
        for (j = 0; j < size2; ++j) {
            if (arr1[i] == arr2[j]) {
                printf("%d\n", arr1[i]);
                break;
            }
        }
    }
}
```

Output

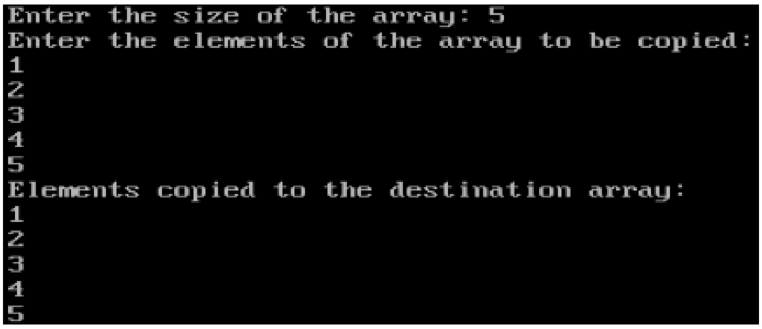


```
Enter the size of the first array: 3
Enter the elements of the first array:
1
2
3
Enter the size of the second array: 4
Enter the elements of the second array:
1
2
3
4
Common elements between the two arrays:
1
2
3
```

Program 39:- WAP to copy all the elements of an array to another.

```
#include <stdio.h>
#include <conio.h>

#define MAX_SIZE 100

void copyArray(int source[], int destination[], int size);

int main()
{
  int sourceArray[MAX_SIZE], destinationArray[MAX_SIZE];
  int size, i;

  clrscr();
  printf("Enter the size of the array: ");
  scanf("%d", &size);

  printf("Enter the elements of the array to be copied:\n");
  for (i = 0; i < size; ++i)
  {
    scanf("%d", &sourceArray[i]);
  }

  copyArray(sourceArray, destinationArray, size);

  printf("Elements copied to the destination array:\n");
  for (i = 0; i < size; ++i)
  {
    printf("%d\n", destinationArray[i]);
  }

  getch();
  return 0;
}

void copyArray(int source[], int destination[], int size)
{
  int i;
  for (i = 0; i < size; ++i)
  {
    destination[i] = source[i];
  }
}
```

Output

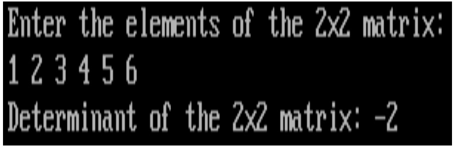


```
Enter the size of the array: 5
Enter the elements of the array to be copied:
1
2
3
4
5
Elements copied to the destination array:
1
2
3
4
5
```

Program 40:- WAP to find determinant of 2x2 Matrix.

```
#include <stdio.h>
#include <conio.h>

int main()
{
   int matrix[2][2];
   int determinant, i, j;

   clrscr();
   printf("Enter the elements of the 2x2 matrix:\n");

   for (i = 0; i < 2; ++i)
   {
     for (j = 0; j < 2; ++j)
     {
        scanf("%d", &matrix[i][j]);
     }
   }

   determinant = matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0];

   printf("Determinant of the 2x2 matrix: %d", determinant);

   getch();
   return 0;
}
```

Output



```
Enter the elements of the 2x2 matrix:
1 2 3 4 5 6
Determinant of the 2x2 matrix: -2
```