EX1

(2 marks, file to be edited: Q6.c)

Your program allows users to enter 5 person names into an array of strings. The program performs sorting of the array in ascending order then prints each element of the array followed by a space character.

Below is an example:

```
John
Joe
Due
Long
Ming

OUTPUT:
Due Joe John Long Ming
Press any key to continue . . .
```

EX2

Your program should allow users to enter an integer 'n'. The program prints hexadecimal representation of 'n' if it is a prime number; else the program prints: "n is not a prime number" where 'n' is the number entered by the user.

Input Format

above

Constraints

above

Output Format

Examples:

```
47

OUTPUT:
0x2F
Press any key to continue . . .
```

```
46

OUTPUT:
46 is not a prime number
Press any key to continue . . .
```

EX3:

(1 mark, file to be edited: Q3.c)

Your program allows users to enter an integer number 'n'.

If 'n' is a palindrome number, the program prints out: *"n is a palindrome number"*

else, the program prints out: *"n is not a palindrome number"*. Here, 'n' is the entered number.

Below is an example of how the program will run:

```
33

OUTPUT:
33 is a palindrome number
Press any key to continue . . .
```

```
127

OUTPUT:
127 is not a palindrome number
Press any key to continue . . .
```

Palindrome: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, 121, 131, 141, 151, 161, 171, 181, 191, 202

1239321

N=123

Loop 1:

123%10=3

re=3

123/10=12

Loop2:

12%10=2

Re=3*10+2=32

12/10=1

Loop3:

1%10=1

Re=32*10+1=321

1/10=0

EX4

Your program allows users to enter array of n integers, where n is entered by the user (n should be kept as a small value, in this case, n <= 10). Your program should then print the sum of squared of each even integer.

Hint: It is possible to use int* array = (int*)malloc(sizeof(int)*n) to create a dynamic array

Below is an example:

```
5
1
2
4
1
3

OUTPUT:
20
Press any key to continue . . .
```

EX5:

Users are required to enter five integer numbers using the keyboard (STDIN). The program needs to find the maximum even number among the entered values. The program then displays this number on screen.

Input Format

Below

Constraints

Below

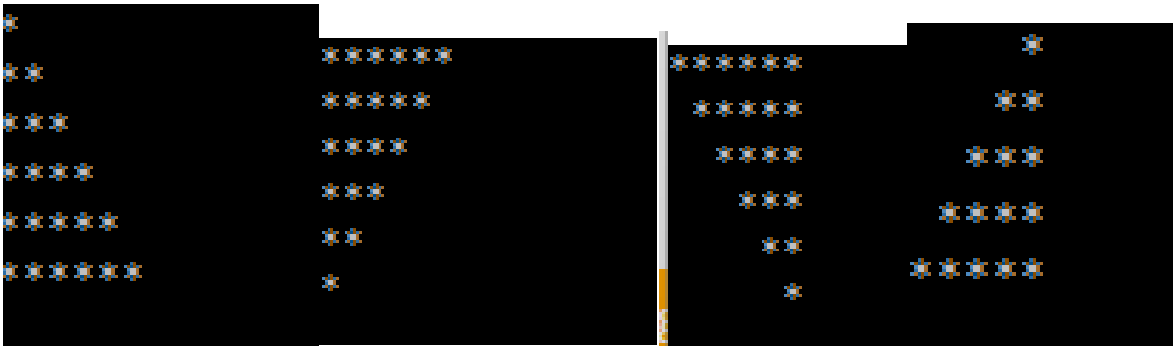
Output Format

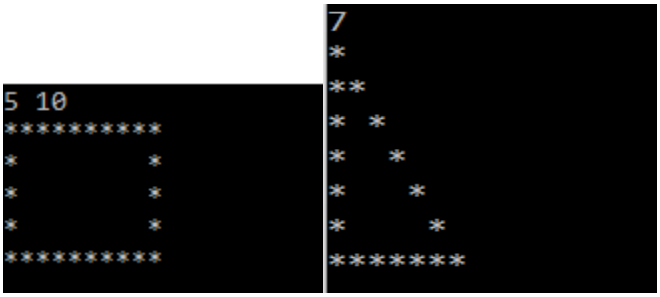
Below is an example of how the program will run:

```
1
7
2
6
4

OUTPUT:
6
Press any key to continue . . .
```

EX6:





EX7:

Your program should allow users to enter an integer number: 'a'. The program should check if 'a' is a power of 2 or not. If it is, the program prints the exponent 'n' that makes the number 'a' the power of 2; else, the program prints: "a is not a power of 2" where 'a' is the entered number from user.

Input Format

above

Constraints

above

Output Format

Example:

```
256

OUTPUT:
8
Press any key to continue . . .
```

```
255

OUTPUT:
255 is not a power of 2
Press any key to continue . . .
```

EX8:

Your program allows users to enter 5 float numbers.

The system displays the entered numbers in descending order.

Below is an example of how the program will run:

```
9.4
8.2
6.3
7.2
5.1

OUTPUT:
9.400000 8.200000 7.200000 6.300000 5.100000
Press any key to continue . . .
```

EX9:

Your program should allow users to enter an integer number 'n', then it should display as follows.

If 'n' is prime number, displays: Not Prime

If 'n' is not prime number, displays: Prime

```
7

OUTPUT:
Not Prime
Press any key to continue . . .
```

```
12

OUTPUT:
Prime
Press any key to continue . . .
```

EX10:

Your program allows users to enter 5 "integer" numbers. The system sorts the entered numbers in ascending order. The system then displays only the even numbers to screen. There is a newline character between any two adjacent numbers.

Input Format

above

Constraints

above

Output Format

Below is an example of how the program will run:

```
3
2
8
6
7

OUTPUT:
2
6
8
Press any key to continue . . .
```

EX11:

Your program allows users to enter array of n integers, where n is entered by the user (n < 10). The program removes all duplicated odd numbers (keeps only the first occurrence of the numbers). Then, the program prints the resultant list of numbers (after removing the duplicated ones). Between any two numbers, there is a newline character.

Input Format

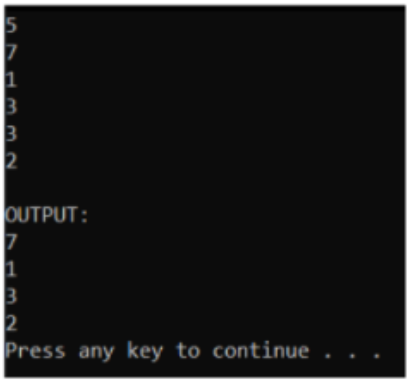
above

Constraints

above

Output Format

Below is an example how the program works.

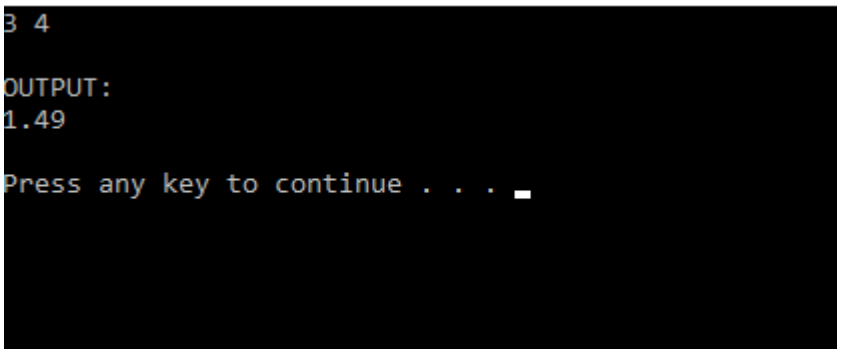


EX12:

Sum = 1+1/x + 1/x^2 + 1/x^3 +... + 1/x^n

Input: x= 3, n=4

Output:



EX13: Write a C program that will print out **sum** of **integers** inputted from the keyboard until the value 0 is inputted.

EX14: Write a C program that will accept two characters then print out ASCII code difference between them and characters between them including code values in decimal, octal, hexadecimal expansions in ascending order.

EX15:

Write a C program that will carry out some times. In each time, a nonnegative integer is accepted then print out the sum of its decimal digits. The program will terminate when its value of accepted number is negative.

EX16:

$S1=1+2+3+4+\ldots..+n$

$S2=1+1/2+1/3+\ldots+1/n$

$S3=x+x^2+x^3+\ldots.+x^n$

$S4=x+x^3+x^5+\ldots.+x^{2n+1}$