



Luxor University

Faculty of computers and information

Programming Fundamentals

Lab Sheet #5

Objectives:

- Learn how to think on paper before coding.
- Learn how to construct your program step by step in detail (Algorithm).
- Learn how to present your program algorithm in an efficient and organized way.
- Start you first C console programs.
- Learn how to solve problems using (array).

Problems:

- Reverse order
- Calculate Average
- Minimum value
- Assign grades
- Print distinct numbers
- Vowels and Consonants
- Duplicate Elimination
- Linear Search
- Sales Commissions

Reverse order

Problem statement:

Write a C program that prints the data of an array in reverse order.

* initialize array using the values: {1,2,3,4,5,6,7,8,9}.

Output:

Reverse order of array.

Example 1:

Output

9 8 7 6 5 4 3 2 1

Calculate Average

Problem statement:

Write a C program that takes the size of array and its elements then return the average of the elements.

Input:

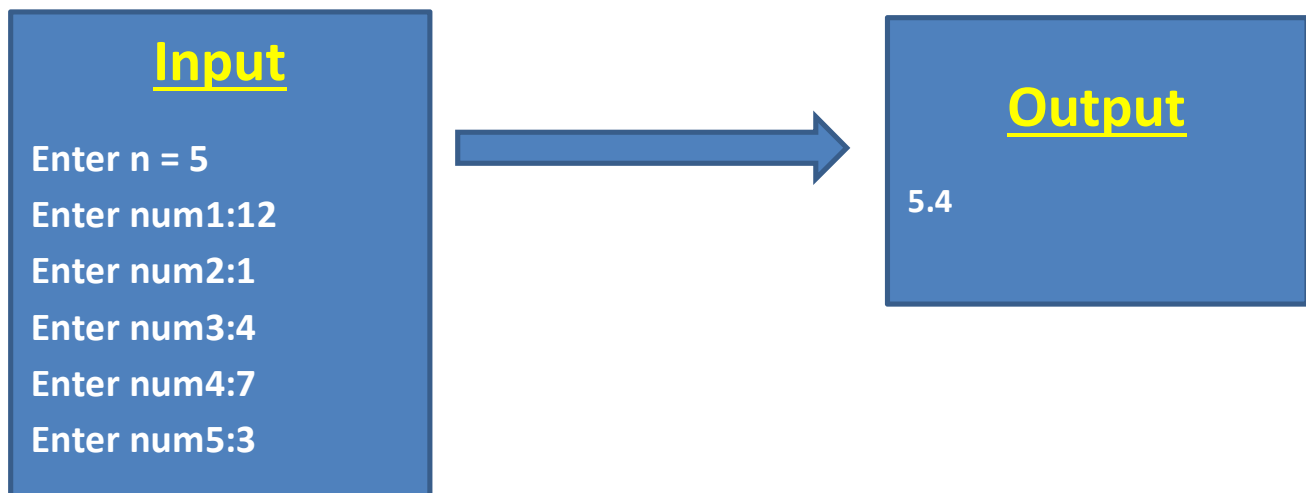
Size of array.

Integer numbers.

Output:

Average

Example 1:



Minimum value

Problem statement:

Write a C program that takes an integer array and the array size is 5 and returns the smallest element of it.

Input:

Array of size 5.

Output:

Minimum element

Example 1:



Example 2:



Assign grades

Problem statement:

Write a program that reads student scores, gets the best score, and then assigns grades based on the following scheme:

- 1) Grade is A if score is $\geq \text{best} - 10$
- 2) Grade is B if score is $\geq \text{best} - 20$;
- 3) Grade is C if score is $\geq \text{best} - 30$;
- 4) Grade is D if score is $\geq \text{best} - 40$;
- 5) Grade is F otherwise.

The program prompts the user to enter the total number of students, then prompts the user to enter all of the scores, and concludes by displaying the grades.

Input:

Number of student.

Grades of the student.

1 integer (best_score)

Output:

displaying the grades(A,B,C,D,F)

Example 1:



Print distinct numbers

Problem statement:

Write a program that reads in 10 numbers and displays distinct numbers (i.e., if a number appears multiple times, it is displayed only once). (Hint: Read a number and store it to an array if it is new. If the number is already in the array, discard it. After the input, the array contains the distinct numbers.)

Input:

Int array with 10 values

Output:

Array with distinct values.

Example 1:

Vowels and Consonants

Problem statement:

Write a program that reads an unspecified number of uppercase or lowercase alphabets, and determines how many of them are vowels and how many are consonants. Enter zero to signify the end of the input.

Input:

array of character

Output:

Number of vowels and consonants char.

Example 1:

Duplicate Elimination

Problem statement:

use a single-subscripted array to solve the following problem.

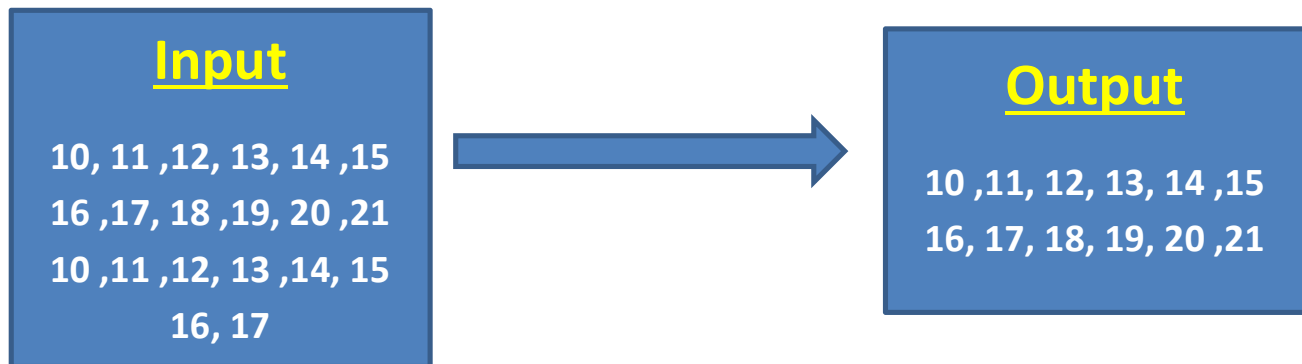
Read in 20 numbers, each of which is between 10 and 100, inclusive. As each number is read, print it only if it's not a duplicate of a number already read.

Input:

Enter 20 integers between 10 and 100

Output:

The no duplicate values

Example 1:**Example 2:**

Linear Search

Problem statement:

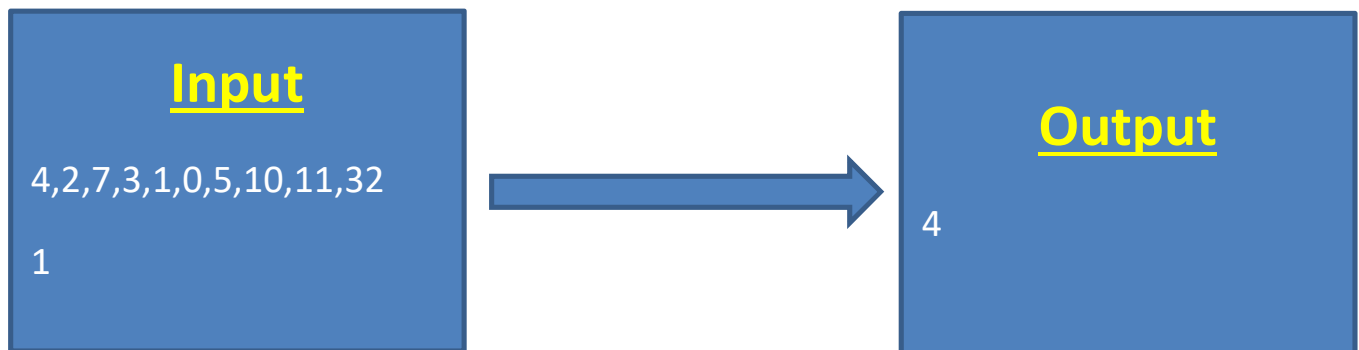
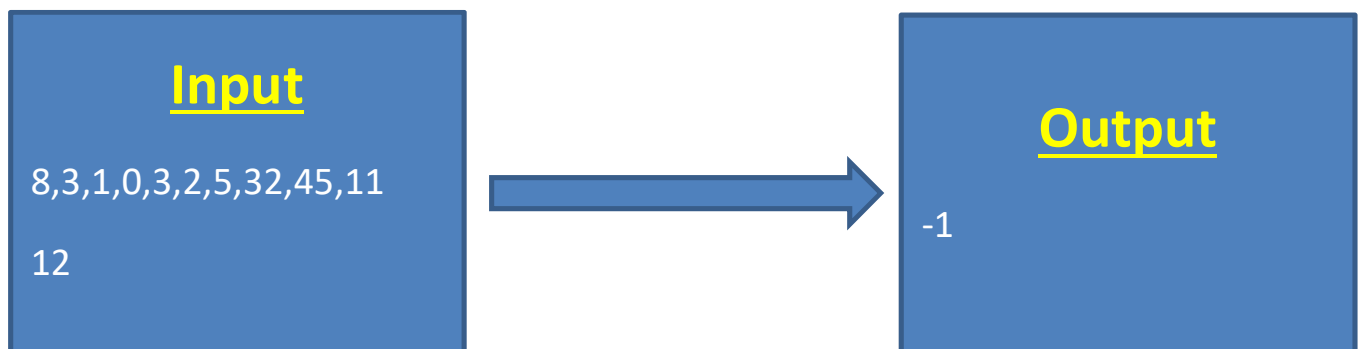
Write a linear Search function to perform the linear search of the array. The function should receive an integer array and the size of the array is 10. If the search key is found, return the Index of the key; otherwise, return -1.

Input:

- Elements of the array
- Search element

Output:

Index of the key or -1 if it not found.

Example 1:**Example 2:**

Sales Commissions

Problem statement:

Use a single-subscripted array to solve the following problem. A company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9% of their gross sales for that week. For example, a salesperson who grosses \$3000 in sales in a week receives \$200 plus 9% of \$3000, or a total of \$470. Write a C program (using an array of counters) that determines how many of the salespeople earned salaries in each of the following ranges (assume that each salesperson's salary is truncated to an integer amount):

- a) \$200–299
- b) \$300–399
- c) \$400–499
- d) \$500–599
- e) \$600–699
- f) \$700–799
- g) \$800–899
- h) \$900–999
- i) \$1000 and over

Input:

1 integers(sales)

Example Run:

```
Enter employee gross sales ( -1 to end ): 3000
Employee Salary is $470.00
Enter employee gross sales ( -1 to end ): 1000
Employee Salary is $290.00
Enter employee gross sales ( -1 to end ): 10000
Employee Salary is $1100.00
Enter employee gross sales ( -1 to end ): 8000
Employee Salary is $920.00
Enter employee gross sales ( -1 to end ): 200
Employee Salary is $218.00
Enter employee gross sales ( -1 to end ): 7000
Employee Salary is $830.00
Enter employee gross sales ( -1 to end ): -1
Employees in the range:
$200-$299 : 2
$300-$399 : 0
$400-$499 : 1
$500-$599 : 0
$600-$699 : 0
$700-$799 : 0
$800-$899 : 1
$900-$999 : 1
Over $1000: 1
```