**Data structures and Algorithms Activity**

**2024-2025 : ODD**

**Section: AF1**

**Course Instructor: Dr.A. Jackulin Mahariba**

**Finding Maximum AND Result from Pairs in an Array of Long Integers**

***Project Title:***

Question 27: Elab Question no.4 in session 8

***Course:***

Data structures and Algorithms

***Team Name:***

VERTEX

***Student Name and Roll Number:***

* **Shreya Nair** RA2311026010251
* **Dayanitha M** RA2311026010288

***Problem Statement***

M is alone and he has an array a1,a2,...,an. M wants to choose two integers i,j such that i≠j,1≤i,j≤n and the value ai&aj (bitwise AND) is maximum.

What is the maximum value M can get?

***Algorithm***

1. Prompt the user to enter the number of elements.
2. Allocate memory for the specified number of long integers.
3. Check if memory allocation succeeded. If it fails, display an error message and exit the program.
4. Prompt the user to input the elements into the allocated memory.
5. Initialize a variable to store the maximum AND result.
6. Use nested loops to iterate through all unique pairs of elements:
   * For each pair, calculate the bitwise AND.
   * If the result is greater than the current maximum, update the maximum.
7. Print the maximum AND result.
8. Free the allocated memory.

***Pseudocode***

FUNCTION main

DECLARE n AS INTEGER

PRINT "Enter the number of elements: "

READ n

ALLOCATE memory for n long integers AS ptr

IF ptr IS NULL THEN

PRINT "Memory allocation failed"

RETURN 1

END IF

CALL input(ptr, n)

FREE ptr

RETURN 0

END FUNCTION

FUNCTION input(ptr, n)

DECLARE i, j AS INTEGER

DECLARE max\_and AS LONG = 0

PRINT "Enter n elements:"

FOR i FROM 0 TO n-1 DO

READ ptr[i]

END FOR

FOR i FROM 0 TO n-1 DO

FOR j FROM i+1 TO n-1 DO

DECLARE temp AS LONG = ptr[i] AND ptr[j]

IF temp > max\_and THEN

max\_and = temp

END IF

END FOR

END FOR

PRINT "The maximum AND result is: ", max\_and

END FUNCTION

***ELAB CODE IN C:***

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void input(long \*, int);

int main()

{ int n;

scanf("%d", &n); // Read number of elements

long \*ptr = (long\*) malloc(n \* sizeof(long)); // Allocate memory

input(ptr, n);

free(ptr); // Free memory

return 0;

}

void input(long \*ptr, int n)

{ int i, j;

int m = 0; // Initialize max AND as 0

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

{ scanf("%ld", ptr + i); } // Read array

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

{ if (\*(ptr + i) <= m)

{ continue; } // Skip if element <= max AND

for (j = i + 1; j < n; j++)

{ int temp = \*(ptr + i) & \*(ptr + j); // Bitwise AND

if(temp > m)

{

m = temp; // Update max AND if needed

}}}

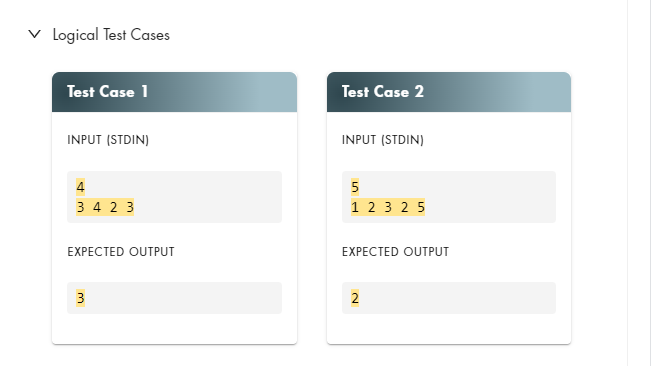
printf("%d", m); } // Output max AND

***OUTPUT:***

A screenshot of a computer

Description automatically generated

***SAMPLE INPUT AND OUTPUT PROVIDED:***

******

***IMPROVED CODE:***

#include <stdio.h>

#include <stdlib.h>

void input(long \*, int);

int main() {

int n;

// Prompt for the number of elements

printf("Enter the number of elements: ");

scanf("%d", &n);

// Allocating memory for n long integers

long \*ptr = (long \*)malloc(n \* sizeof(long));

// Check if malloc succeeded

if (ptr == NULL) {

printf("Memory allocation failed\n");

return 1;

}

// Taking input

input(ptr, n);

// Freeing the allocated memory

free(ptr);

return 0;

}

void input(long \*ptr, int n) {

int i, j;

long max\_and = 0; // To store the maximum AND result

// Prompt for array elements

printf("Enter %d elements:\n", n);

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

scanf("%ld", &ptr[i]);

}

// Find maximum AND result from pairs

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

for (j = i + 1; j < n; j++) {

long temp = ptr[i] & ptr[j];

// Update max\_and if temp is greater

if (temp > max\_and) {

max\_and = temp; // Update max\_and if a new max is found

}

}

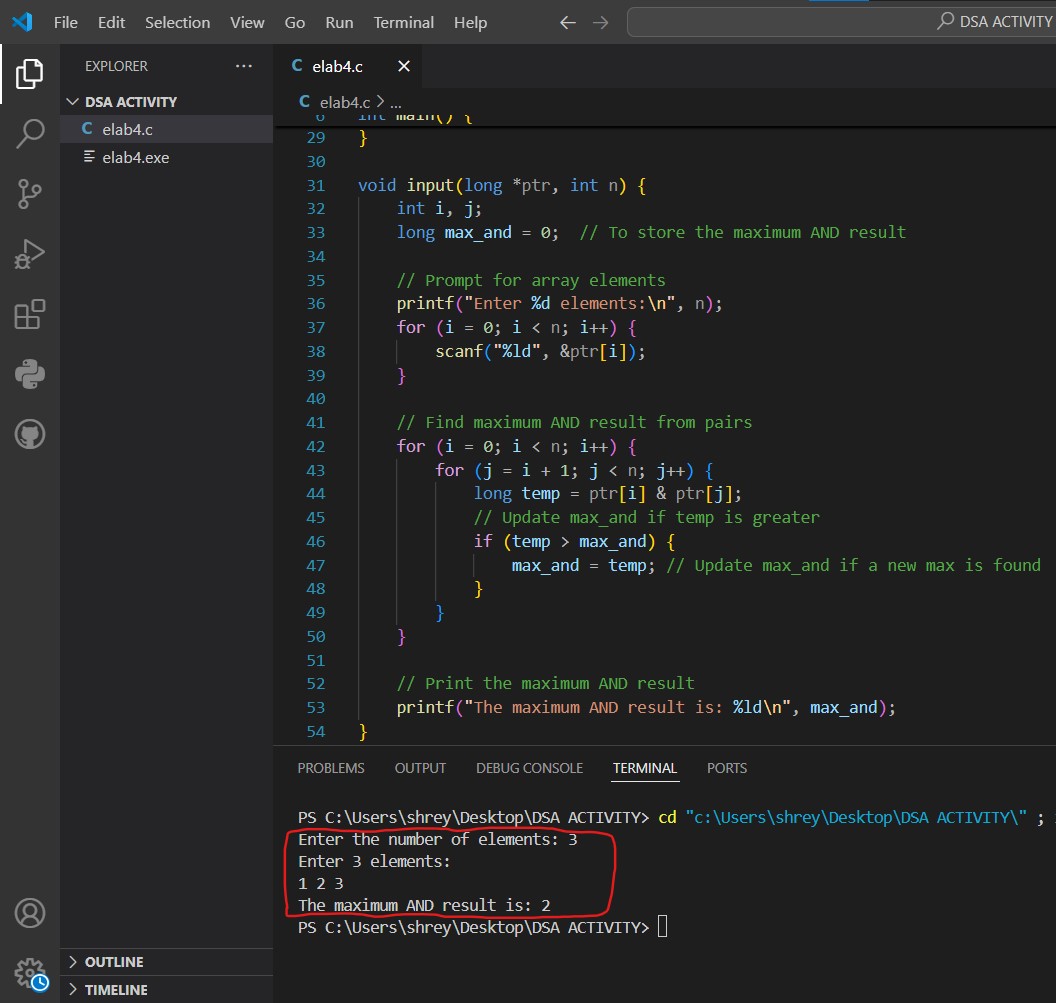
}

// Print the maximum AND result

printf("The maximum AND result is: %ld\n", max\_and);

}

***OUTPUT:***

******

***RESULT:***

The program successfully computes the maximum bitwise AND result from unique pairs of long integers entered by the user. For example, given the input numbers **1,2,3** it finds that the maximum AND result is **2** from the pair **(2,3)**. This demonstrates effective memory management and bitwise operations. The objectives of the program have been achieved.

***GITHUB LINK+VIDEO:***

<https://drive.google.com/file/d/1FHTnK2Sr8hOuEKCB82fdvRJnBpBjFbz5/view?usp=sharing>