

Algorithm: Addition Using Pointers

Step 1: Start

Step 2: Declare variables

- Declare three integer variables: num1, num2, and sum.
- Declare two integer pointers: ptr1, ptr2.

Step 3: Display program title

Print "**** Addition using Pointers ****" on the screen.

Step 4: Input two numbers

- Prompt the user to enter the **first number**, and store it in num1.
- Prompt the user to enter the **second number**, and store it in num2.

Step 5: Assign addresses to pointers

- Set ptr1 = &num1 (pointer ptr1 now points to num1).
- Set ptr2 = &num2 (pointer ptr2 now points to num2).

Step 6: Perform addition using pointers

- Access the values stored at the addresses of ptr1 and ptr2 using the dereference operator *.
- Compute the sum:
$$\text{sum} = *ptr1 + *ptr2.$$

Step 7: Display the result

Print the message:

"Sum of num1 and num2 is: sum".

Step 8: Stop

Pseudocode

Step 1: START

Step 2: DECLARE num1, num2, sum AS INTEGER

Step 3: DECLARE ptr1, ptr2 AS POINTER TO INTEGER

Step 4: PRINT "** Addition using Pointers ****"**

Step 5: PRINT "Enter first number: "

Step 6: READ num1

Step 7: PRINT "Enter second number: "

Step 8: READ num2

Step 9: SET ptr1 = ADDRESS OF num1

Step 10: SET ptr2 = ADDRESS OF num2

Step 11: SET sum = VALUE AT ptr1 + VALUE AT ptr2

Step 12: PRINT "Sum of", num1, "and", num2, "is:", sum

Step 13: STOP

Algorithm: Array Reverser Using Pointers

Step 1: Start

Step 2: Declare variables

- Declare an integer variable n (to store the size of the array).
- Declare an integer array arr[n].
- Declare a pointer ptr and assign it to point to the base address of the array:
ptr = arr.

Step 3: Display program title

Print: "*** Array Reverser ***".

Step 4: Input array size

Prompt the user:

"Enter the size of the array:"

Read the value of n.

Step 5: Input array elements using pointer

1. Print: "Enter the elements of the array:"
2. For each index i from 0 to n - 1:
 - a. Display the index number i.
 - b. Read an integer value using:
scanf("%d", ptr + i);
(This stores the value directly into the correct memory location using the pointer.)

Step 6: Display array elements in reverse order

1. Print: "Array elements in reverse order:"
2. For each index i from n - 1 down to 0:
 - a. Print the element using pointer arithmetic:
*(ptr + i).

Step 7: Stop

Pseudocode

Step 1: START

Step 2: DECLARE n AS INTEGER

Step 3: PRINT "* Array Reverser ***"**

Step 4: PRINT "Enter the size of the array:"
READ n

Step 5: DECLARE arr[n] AS INTEGER

Step 6: SET ptr = arr

Step 7: PRINT "Enter the elements of the array:"
FOR i = 0 TO n - 1 DO
PRINT i, ":"
READ VALUE INTO (ptr + i)
END FOR

Step 8: PRINT "Array elements in reverse order:"

Step 7: FOR i = n - 1 DOWNTO 0 DO
PRINT VALUE AT (ptr + i)
END FOR

Step 8: STOP