**Group- Gerald, Carlo, Alexis**

**Question--Write an algorithm to add two numbers entered by a user.**

Algorithm to add two numbers entered by the user

1. Start
2. Initiate variables; X, Y and SUM
3. Prompt the user to enter the first number, X
4. Read the first number and store it in variable X
5. Prompt the user to enter the second number Y
6. Read the second number and store it in variable Y
7. Add the two numbers and store it in SUM; SUM=X+Y
8. Display SUM of the two numbers
9. End.

**Question---Write an algorithm and pseudo code of calculating the sum of n elements of an array**.

**Algorithm**

1. Start
2. Initialize n, i, SUM=0
3. Prompt the user to enter the value of n
4. Read the number entered store it in the variable n
5. Declare and array of size n
6. Iterate through the first n elements of the array
7. Add each element to sum
8. Display the value of sum
9. End

**Pseudocode**

1. Start
2. PRINT “enter the number of elements n”
3. READ n, i,
4. Declare array[n]
5. Declare SUM=0
6. PRINT “enter”, n, ”elements”
7. FOR i=0 to n-1:
   1. READ array[i]
   2. SUM=SUM + array[i]
8. End FOR
9. PRINT “Sum of array of elements”, SUM
10. End

**Question--Write a c code that shows how to insert an element entered by the user into the middle of the array.**

#include <stdio.h>

int main() {

int array[100];

int n, i, newElement, position;

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

scanf("%d", &n);

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

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

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

}

printf("Enter the new element to insert: ");

scanf("%d", &newElement);

position = n / 2;

for (i = n; i > position; i--) {

array[i] = array[i - 1];

}

array[position] = newElement;

n++;

printf("Array after inserting the new element:\n");

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

printf("%d ", array[i]);

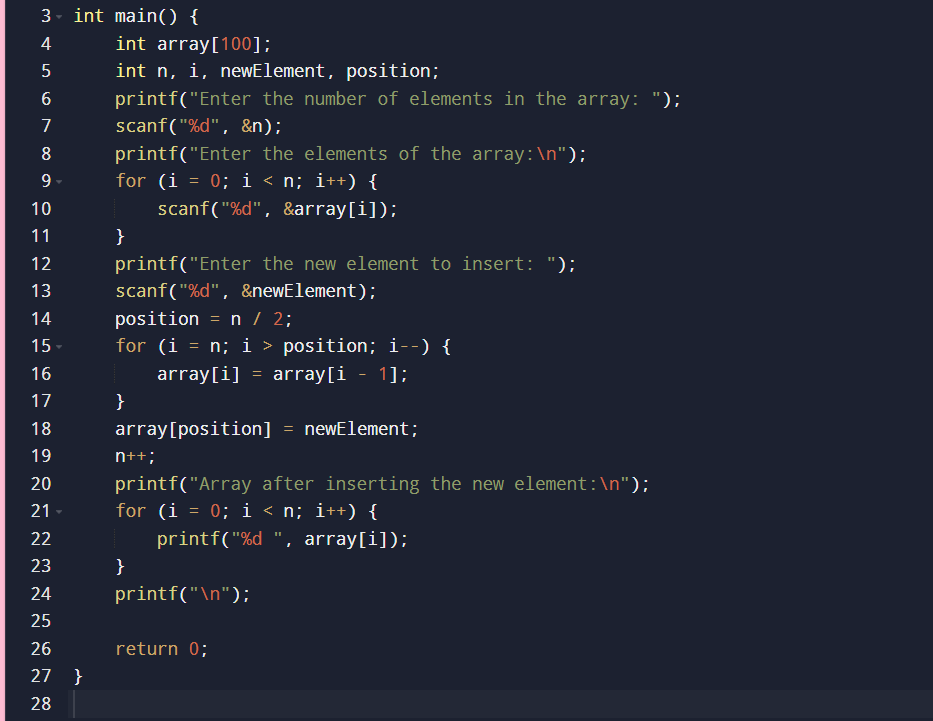
}

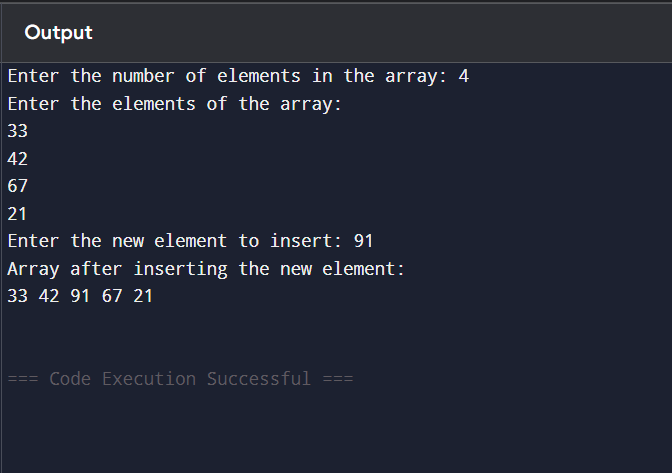
printf("\n");

return 0;

}

**Screenshot of output**

****

**OUTPUT**

**ASSIGNMENT ON STACKS (ASSIGNMENT 2)**

**This is what I wrote**

**1. Write a C/C++ code showing how to push an element into a stack.**

**2. Using friends, write a C/C++ code showing how to peek a stack.**

**#include <stdio.h>**

**#include <string.h>**

**#define MAX 5 // Stack size**

**#define NAME\_SIZE 50 // Maximum length for each name**

**char stack[MAX][NAME\_SIZE]; // Stack to store names**

**int top = -1; // Stack top index**

**// Push function (adds name to stack)**

**void push() {**

**if (top == MAX - 1) {**

**printf("Stack Overflow\n");**

**} else {**

**top++;**

**printf("Enter a friend's name:");**

**scanf(" %49[^\n]", stack[top]); // Reads a string with spaces**

**}**

**}**

**// Pop function (removes name from stack)**

**void pop() {**

**if (top == -1) {**

**printf("Stack Underflow\n");**

**} else {**

**printf("Popped: %s\n", stack[top]);**

**top--;**

**}**

**}**

**// Peek function (displays the top element)**

**void peek() {**

**if (top == -1) {**

**printf("Stack is empty\n");**

**} else {**

**printf("Top Friend: %s\n", stack[top]);**

**}**

**}**

**// Print the entire stack**

**void printStack() {**

**if (top == -1) {**

**printf("Stack is empty\n");**

**return;**

**}**

**printf("\nStack (Top-> Bottom):\n");**

**for (int i = top; i >= 0; i--) {**

**printf("|%s|\n ", stack[i]);**

**}**

**printf("\n");**

**}**

**int main() {**

**int choice;**

**do {**

**printf("\nStack Operations:\n");**

**printf("1. Push (Add Friend)\n");**

**printf("2. Pop (Remove Friend)\n");**

**printf("3. Peek (View Top Friend)\n");**

**printf("4. Print Stack\n");**

**printf("5. Exit\n");**

**printf("Enter your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1:**

**push();**

**break;**

**case 2:**

**pop();**

**break;**

**case 3:**

**peek();**

**break;**

**case 4:**

**printStack();**

**break;**

**case 5:**

**printf("Exiting...\n");**

**break;**

**default:**

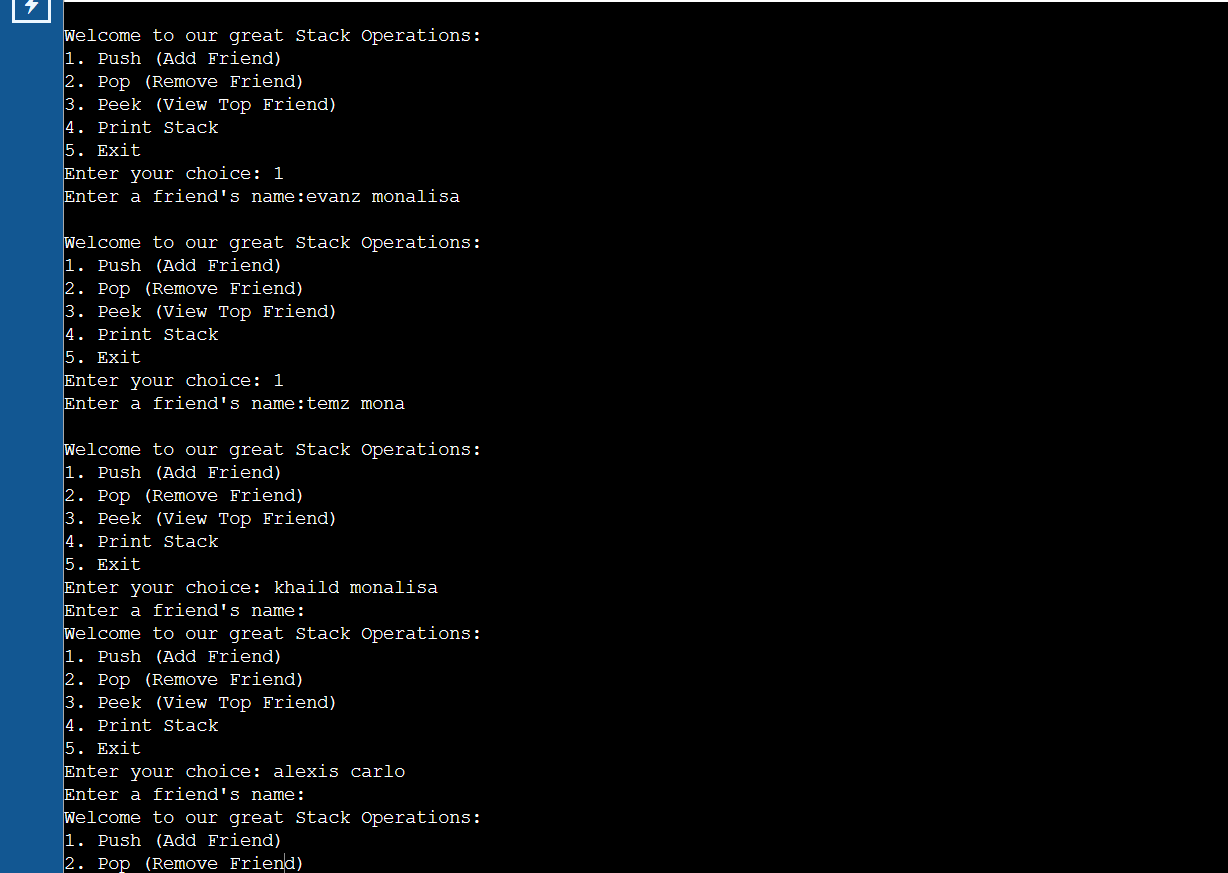
**printf("Invalid choice! Try again.\n");**

**}**

**} while (choice != 5);**

**return 0;**

**OUT PUT FOR THE CODE**

****