Bachelor of Information Technology (Hons)

Assignment Cover Sheet

Course Code:__EC3105_________________ Course Title:__C Programming______________________

Assignment Title: __Practical_________________________ Due Date: __26th April 2023_________________

Date Submitted: _26th April 2023___________ Lecturer Name: _Aashish Acharya______________

To be completed if this is an individual assignment

I declare that this assignment is my individual work. I have not worked collaboratively nor have I copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

StudentName:_Amrita Tamang___ Student ID:_00020709_____________ Signature: __________

To be completed if this is a group assignment

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

Student ID	Student Name	Signature
__________	____	___________
__________	____	___________
__________	____	___________
__________	____	___________

Lecturer's comments: ____

Total Marks: _______________ Lecturer's Signature:__________________

Feedback to Student:

I/We acknowledged receiving feedback from the lecturer on this assignment.

Student's Signature: _____________ ______________ ______________ ____________

Extension certification:

This assignment has been given an extension and is now due on ________________.

Lecturer's Signature: ______________

Contents

1. Write a simple program to create a mini calculator that performs simple operations of addition, subtraction, multiplication and division. Your program should be menu driven, should ask for appropriate user input and produce the result. Make use of correct data types and proper formatting in the output.

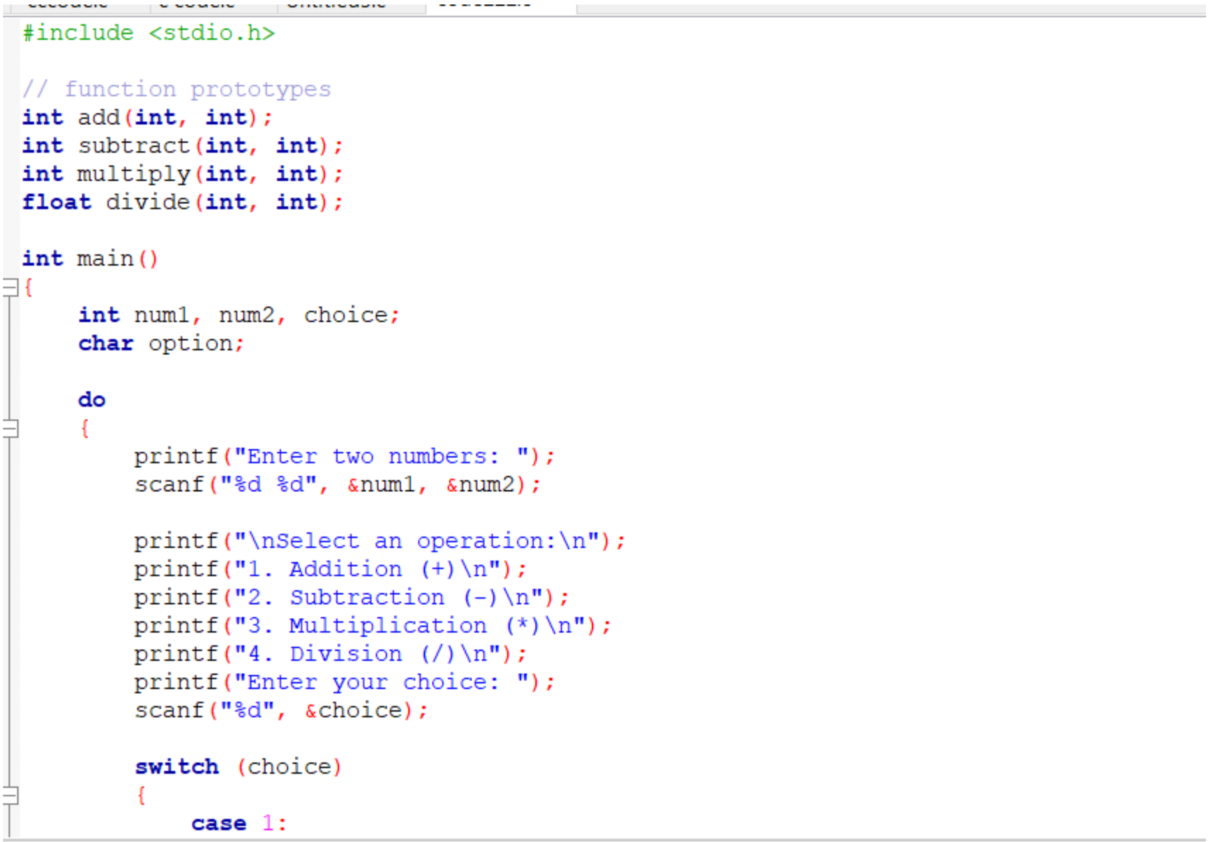
For each operation that your system performs, create a user defined function to carry out the operations. Design a menu driven system that gives user a choice of operations when the system begins execution.

- **Code copy**

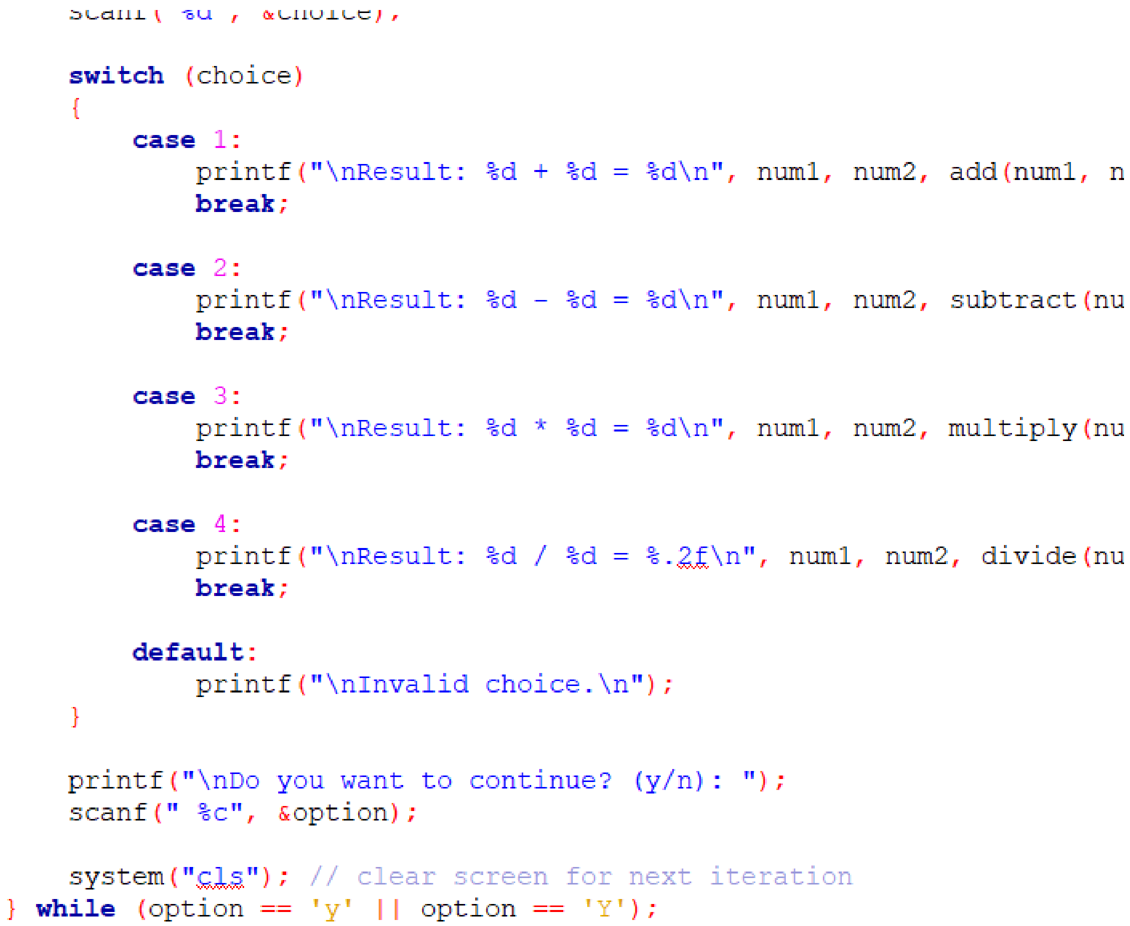
```
#include <stdio.h>
// function prototypes
int add(int, int);
int subtract(int, int);
int multiply(int, int);
float divide(int, int);
int main()
{
    int num1, num2, choice;
    char option;
    do
    {
        printf("Enter two numbers: ");
        scanf("%d %d", &num1, &num2);

        printf("\nSelect an operation:\n");
        printf("1. Addition (+)\n");
        printf("2. Subtraction (-)\n");
        printf("3. Multiplication (*)\n");
        printf("4. Division (/)\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
    switch (choice)
    {
        case 1:
            printf("\nResult: %d + %d = %d\n", num1, num2, add(num1, num2));
            break;

        case 2:
            printf("\nResult: %d - %d = %d\n", num1, num2, subtract(num1, num2));
            break;

        case 3:
            printf("\nResult: %d * %d = %d\n", num1, num2, multiply(num1, num2));
            break;

        case 4:
            printf("\nResult: %d / %d = %.2f\n", num1, num2, divide(num1, num2));
            break;

         default:
            printf("\nInvalid choice.\n");
    }
           printf("\nDo you want to continue? (y/n): ");
    scanf(" %c", &option);

    system("cls"); // clear screen for next iteration
} while (option == 'y' || option == 'Y');

return 0;
```

```
}

// function definitions
int add(int x, int y)
{
    return x + y;
}

int subtract(int x, int y)
{
    return x - y;
}

int multiply(int x, int y)
{
    return x * y;
}

float divide(int x, int y)
{
    return (float)x / y;
}
```

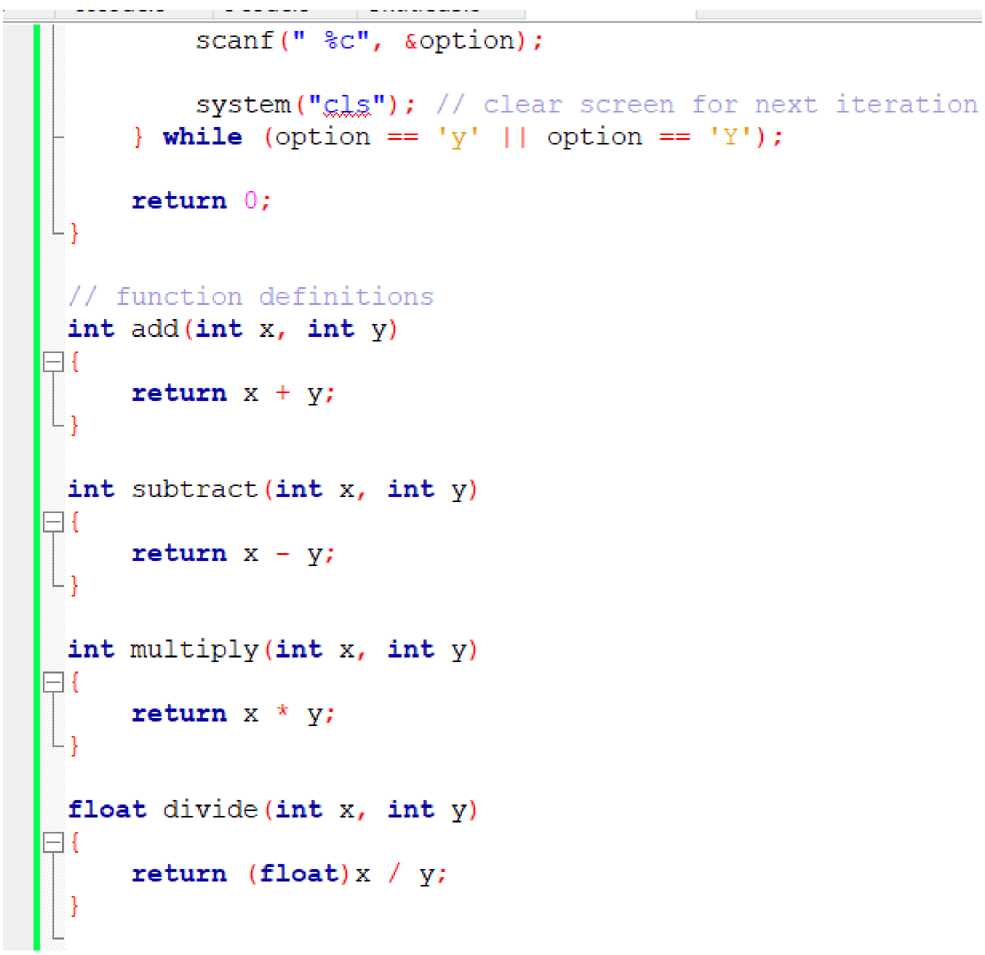
- **Code Screenshot**



```
#include <stdio.h>

// function prototypes
int add(int, int);
int subtract(int, int);
int multiply(int, int);
float divide(int, int);

int main()
{
    int num1, num2, choice;
    char option;

    do
    {
        printf("Enter two numbers: ");
        scanf("%d %d", &num1, &num2);

        printf("\nSelect an operation:\n");
        printf("1. Addition (+)\n");
        printf("2. Subtraction (-)\n");
        printf("3. Multiplication (*)\n");
        printf("4. Division (/)\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
```

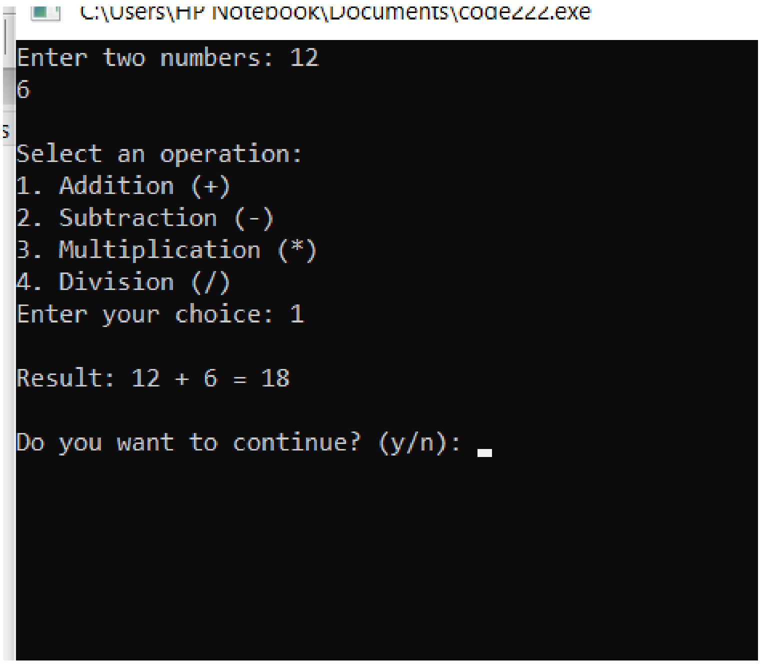


```
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
                printf("\nResult: %d + %d = %d\n", num1, num2, add(num1, n
                break;

            case 2:
                printf("\nResult: %d - %d = %d\n", num1, num2, subtract(nu
                break;

            case 3:
                printf("\nResult: %d * %d = %d\n", num1, num2, multiply(nu
                break;

            case 4:
                printf("\nResult: %d / %d = %.2f\n", num1, num2, divide(nu
                break;

            default:
                printf("\nInvalid choice.\n");
        }

        printf("\nDo you want to continue? (y/n): ");
        scanf(" %c", &option);

        system("cls"); // clear screen for next iteration
    } while (option == 'y' || option == 'Y');
```

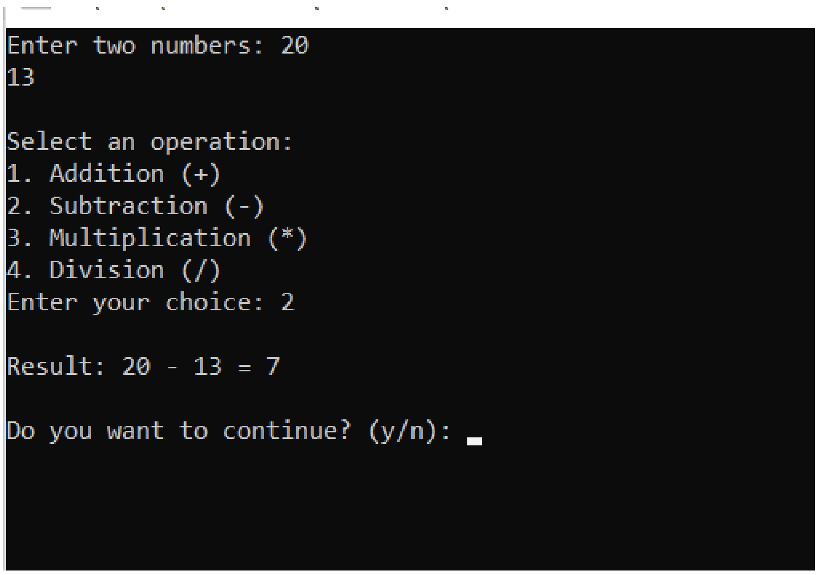


```
        scanf(" %c", &option);

        system("cls"); // clear screen for next iteration
    } while (option == 'y' || option == 'Y');

    return 0;
}

// function definitions
int add(int x, int y)
{
    return x + y;
}

int subtract(int x, int y)
{
    return x - y;
}

int multiply(int x, int y)
{
    return x * y;
}

float divide(int x, int y)
{
    return (float)x / y;
}
```

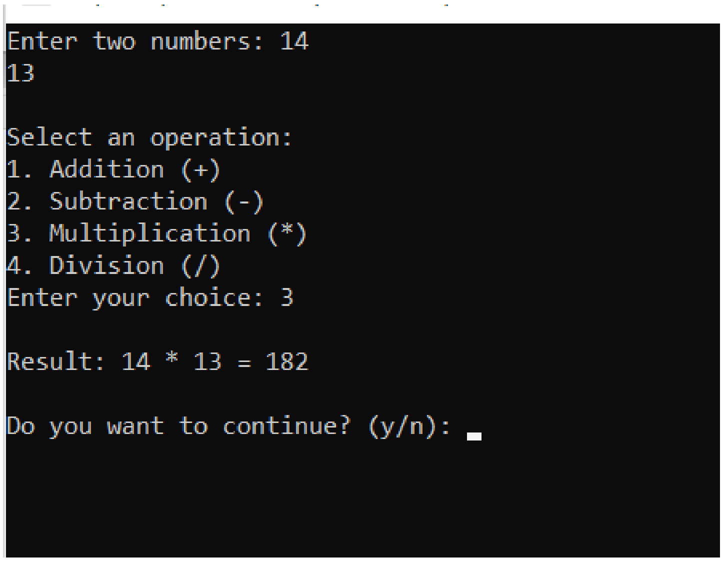
- **Output screenshot**



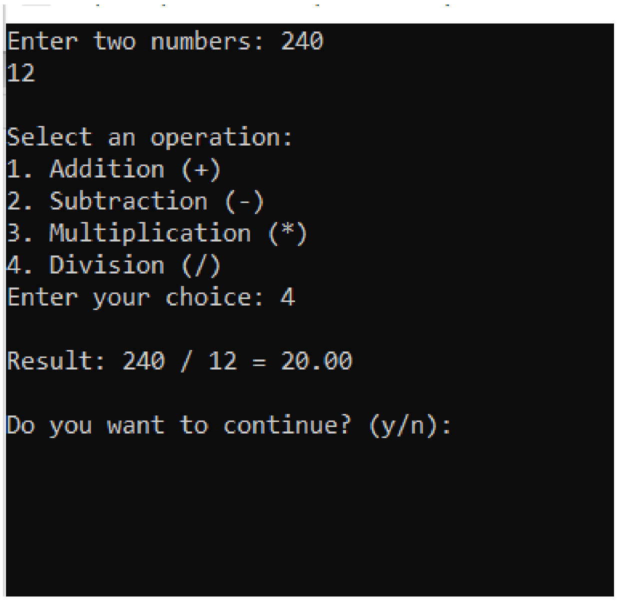
```
Enter two numbers: 12
6

Select an operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter your choice: 1

Result: 12 + 6 = 18

Do you want to continue? (y/n):
```



```
Enter two numbers: 20
13

Select an operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter your choice: 2

Result: 20 - 13 = 7

Do you want to continue? (y/n):
```



```
Enter two numbers: 14
13

Select an operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter your choice: 3

Result: 14 * 13 = 182

Do you want to continue? (y/n):
```



```
Enter two numbers: 240
12

Select an operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter your choice: 4

Result: 240 / 12 = 20.00

Do you want to continue? (y/n):
```

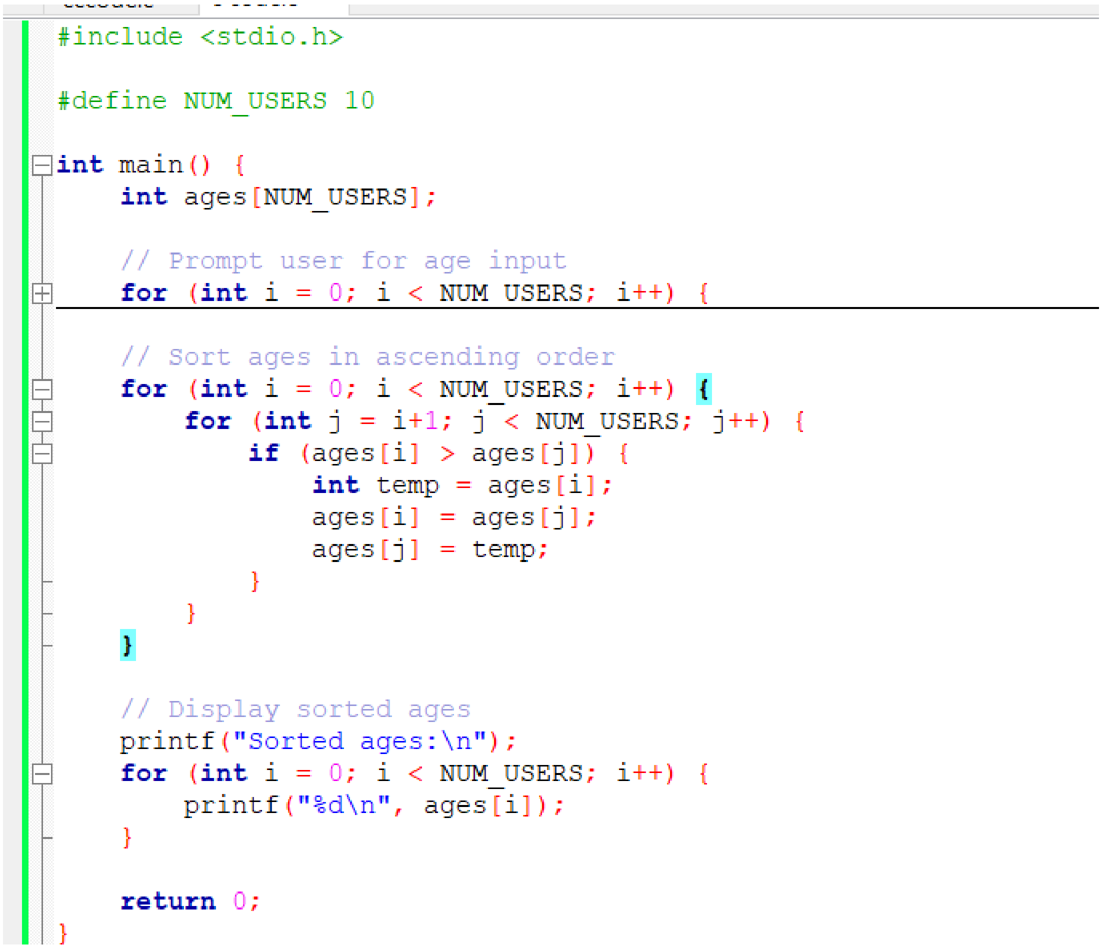
2. Write a program to read the age from a user. Your program should be able to work for 10 consecutive users at a time. Create suitable array implementation to store user input. Once the input is taken, your program should sort the user's age in the ascending order, showing the youngest age on the top of the list and the oldest age at the last. The final result needs to be stored in a separate array before displaying the result.

- **Code copy**

```
#include <stdio.h>


#define NUM_USERS 10


int main() {
   int ages[NUM_USERS];


   // Prompt user for age input
   for (int i = 0; i < NUM_USERS; i++) {
      printf("Enter age for user %d: ", i+1);
      scanf("%d", &ages[i]);
   }


   // Sort ages in ascending order
   for (int i = 0; i < NUM_USERS; i++) {
      for (int j = i+1; j < NUM_USERS; j++) {
         if (ages[i] > ages[j]) {
            int temp = ages[i];
            ages[i] = ages[j];
            ages[j] = temp;
         }
```

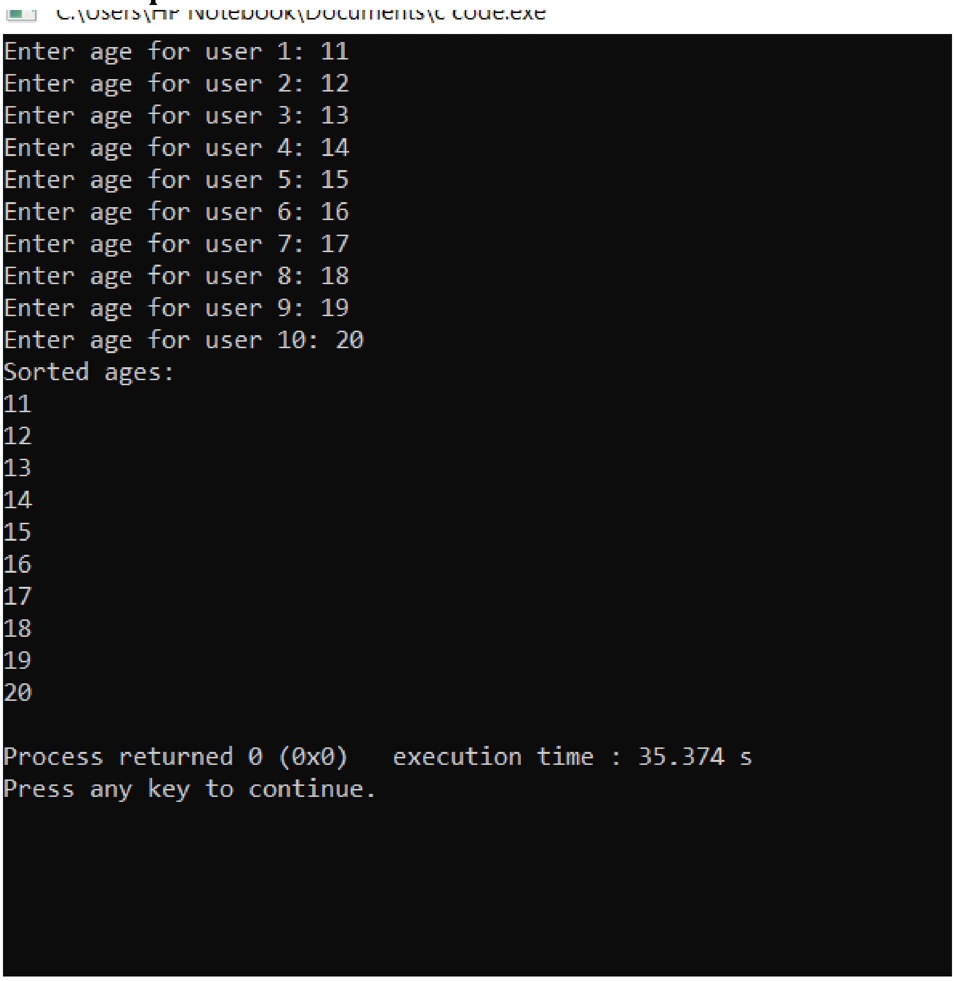
```
        }
    }

    // Display sorted ages
    printf("Sorted ages:\n");
    for (int i = 0; i < NUM_USERS; i++) {
        printf("%d\n", ages[i]);
    }

    return 0;
}
```

- **Code screenshot**



```
#include <stdio.h>

#define NUM_USERS 10

int main() {
    int ages[NUM_USERS];

    // Prompt user for age input
    for (int i = 0; i < NUM_USERS; i++) {

    // Sort ages in ascending order
    for (int i = 0; i < NUM_USERS; i++) {
        for (int j = i+1; j < NUM_USERS; j++) {
            if (ages[i] > ages[j]) {
                int temp = ages[i];
                ages[i] = ages[j];
                ages[j] = temp;
            }
        }
    }

    // Display sorted ages
    printf("Sorted ages:\n");
    for (int i = 0; i < NUM_USERS; i++) {
        printf("%d\n", ages[i]);
    }

    return 0;
}
```

- **Output**



```
Enter age for user 1: 11
Enter age for user 2: 12
Enter age for user 3: 13
Enter age for user 4: 14
Enter age for user 5: 15
Enter age for user 6: 16
Enter age for user 7: 17
Enter age for user 8: 18
Enter age for user 9: 19
Enter age for user 10: 20
Sorted ages:
11
12
13
14
15
16
17
18
19
20

Process returned 0 (0x0)   execution time : 35.374 s
Press any key to continue.
```

Marking Scheme

Course: EC3105 (C Programming)

Lecturer: Mr. Aashish Acharya

Student name: _Amrita Tamang________

criteria	Marks Allocated	Marks Obtained	comments
- Correct Outcome - Screenshot and Code -Output with correct data - Output with faulty data	40 (20 marks each)		
Demonstration and Viva	10 (5 marks each)		

THE END…