PROGRAMMING

CT103

Week 4b

DISC

- Website: https://www.universityofgalway.ie/science-engineering/school-of-computer-science/currentstudents/computerdisc/

- What services does ComputerDISC provide to students?
  - One-to-one advice and support for students
  - Books, courseware, web links, and other learning resources for programming students
  - A website with information and an email service for all queries
  - Advice for students who wish to learn new programming languages autonomously
  - Assistance with new technologies for project work such as Final Year Projects

Lecture Content

- Last lecture (Week 4a):
  - Recap on while loops.
  - For loops.
  - Example C program.

- Today’s evening lecture (Week 4b):
  - Arrays
  - Arrays and loops
  - Arrays example C program

ARRAYS

Array Definition

- What is an **array**?

- **Definition**: An array is a data structure consisting of a collection of elements. Each element can be identified by an index.

Array Definition

- What are **arrays** used for?

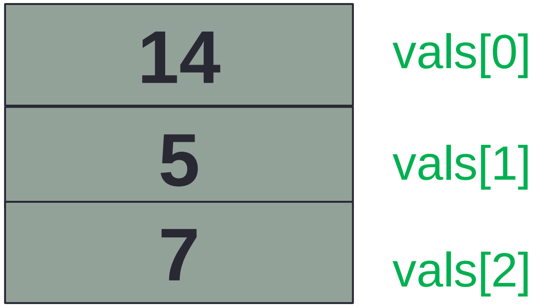
- An array is used to store a collection of data.

- You can think of an array as a collection of variables of the same type.

Arrays in C

- You can define arrays of any type as follows:
    - E.g. int vals [3];
    - E.g. char initials[3];

- You can initialise like this if you want to:
    - int vals[3] = {14,5,7};



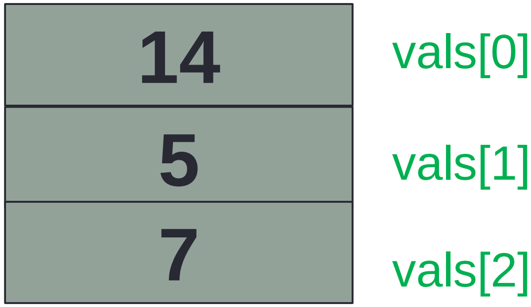
14 vals[0]

5 vals[1]

7 vals[2]

Arrays in C

- If we have the following array called **vals**.
- The size of vals is 3.
- The 1st element is at position 0 of the array.
- The 2nd element is at position 1 of the array.
- The 3rd element is at position 2 of the array.



14 vals[0]

5 vals[1]

7 vals[2]

Array Terminology

- Array – a collection of data.
- Element – one of the "*items*" in the array.
- Index – the position of the element in the array.
- Array size – how many elements in the array.

Initializing an Array

- float prices[3] = {65.56, 45.63, 7.90};

- double salary[2] = {45000.00, 33500.00};

- int grades[5] = {44, 55, 66, 33, 88};

Initializing an Array

- int ages[5] = {6,8,9,11,14}; /* Correct */
- int ages[]; /* Incorrect */
- int ages[] = {6,8,9,11,14}; /* Correct */

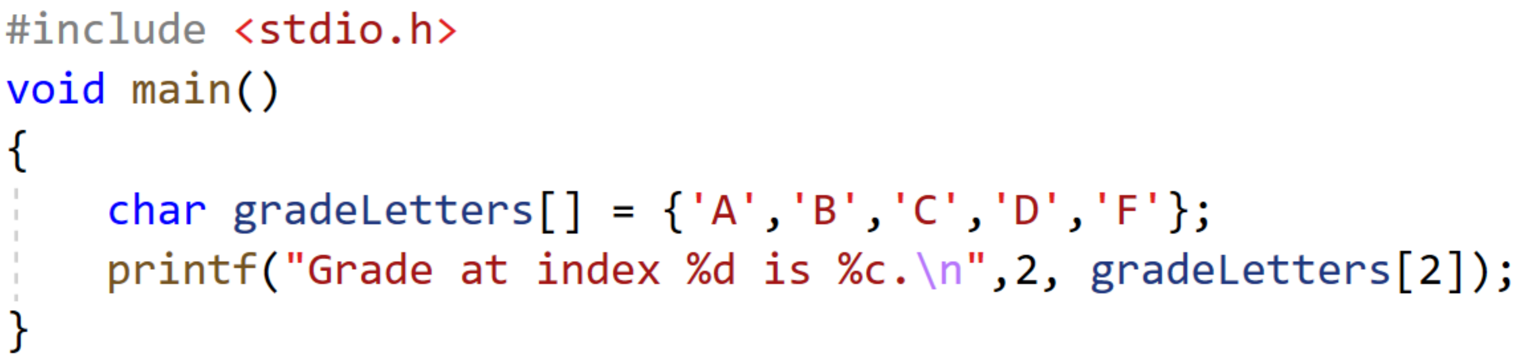
- Remember, you must initialise you array properly.

Simple Array Problem Example

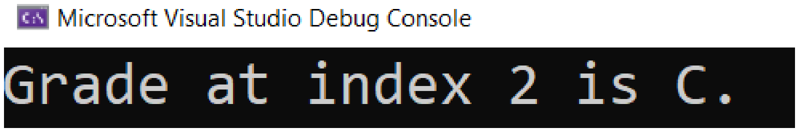
- Write a program that does the following:
- Store all of the possible letter grades that a student can get in an array.
- Print the grade at index 2 to the screen.

Simple Array Problem Example

- The following program creates at array for grade letters.



```
#include <stdio.h>
void main()
{
    char gradeLetters[] = {'A','B','C','D','F'};
    printf("Grade at index %d is %c.\n",2, gradeLetters[2]);
}
```

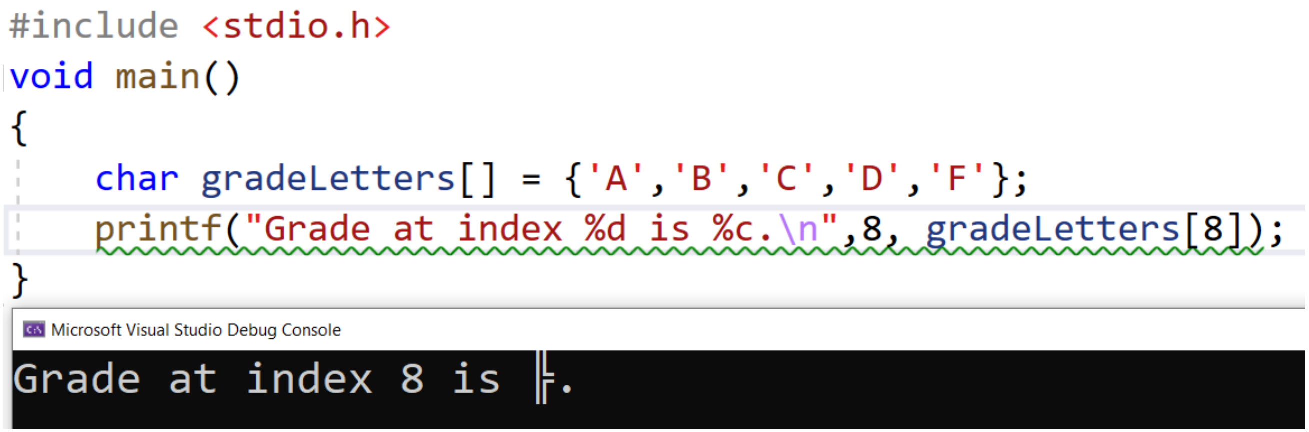


Microsoft Visual Studio Debug Console

```
Grade at index 2 is C.
```

Common Array Mistakes

- Accessing an index that is equal to or larger than the size of the array.
- Don’t do this.

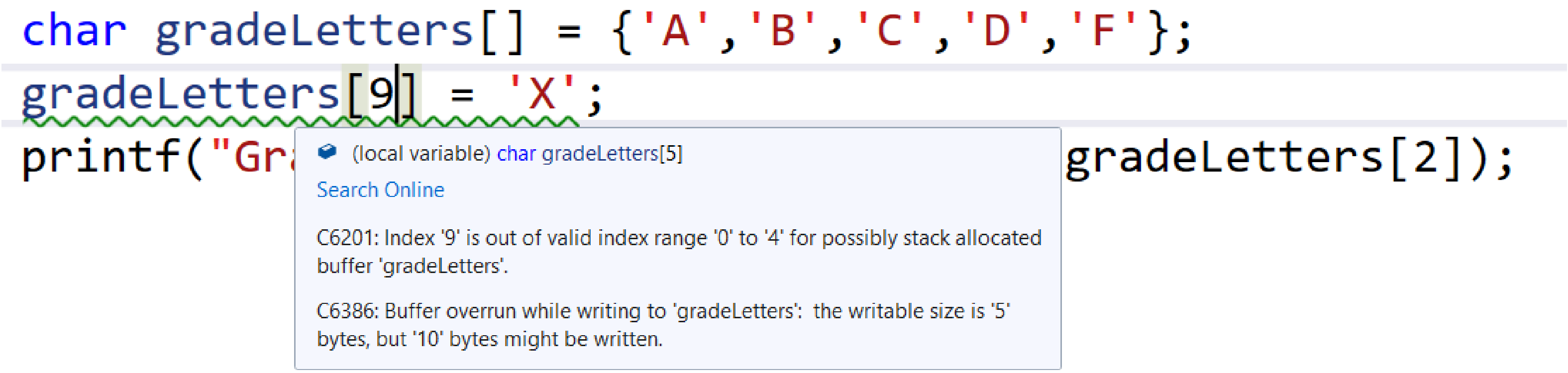


```c
#include <stdio.h>
void main()
{
    char gradeLetters[] = {'A','B','C','D','F'};
    printf("Grade at index %d is %c.\n",8, gradeLetters[8]);
}
```

```
Grade at index 8 is ╟.
```

Common Array Mistakes

- Setting the value of an array element who's index is equal to or larger than the size of the array.
- Don’t do this either.



```
char gradeLetters[] = {'A','B','C','D','F'};
gradeLetters[9] = 'X';
printf("Gr                                       gradeLetters[2]);
```

(local variable) char gradeLetters[5]

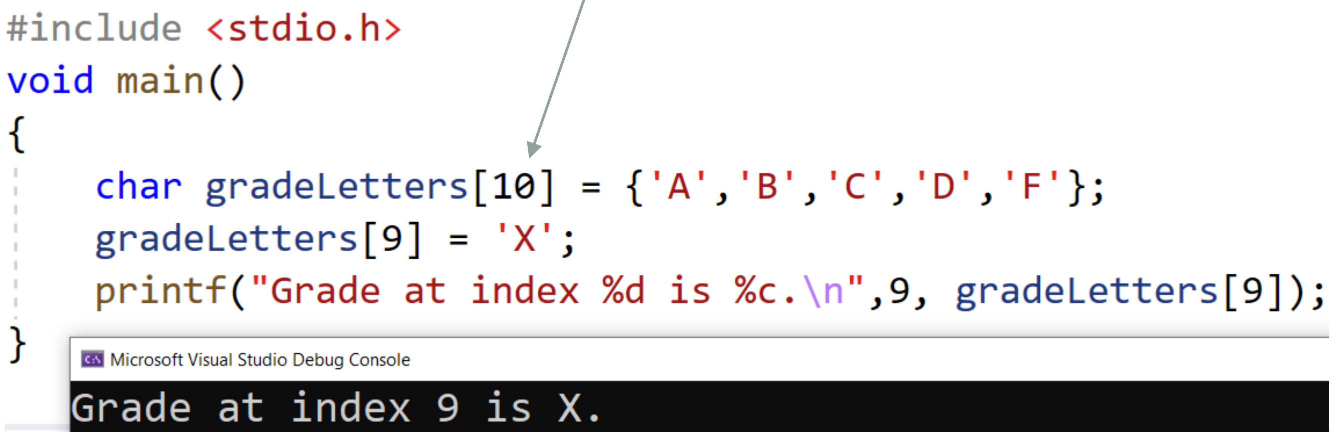
Search Online

C6201: Index '9' is out of valid index range '0' to '4' for possibly stack allocated buffer 'gradeLetters'.

C6386: Buffer overrun while writing to 'gradeLetters': the writable size is '5' bytes, but '10' bytes might be written.

Common Array Mistakes

- I would need to create a larger array.



```
#include <stdio.h>
void main()
{
    char gradeLetters[10] = {'A','B','C','D','F'};
    gradeLetters[9] = 'X';
    printf("Grade at index %d is %c.\n",9, gradeLetters[9]);
}
```

```
Grade at index 9 is X.
```

ARRAYS AND LOOPS

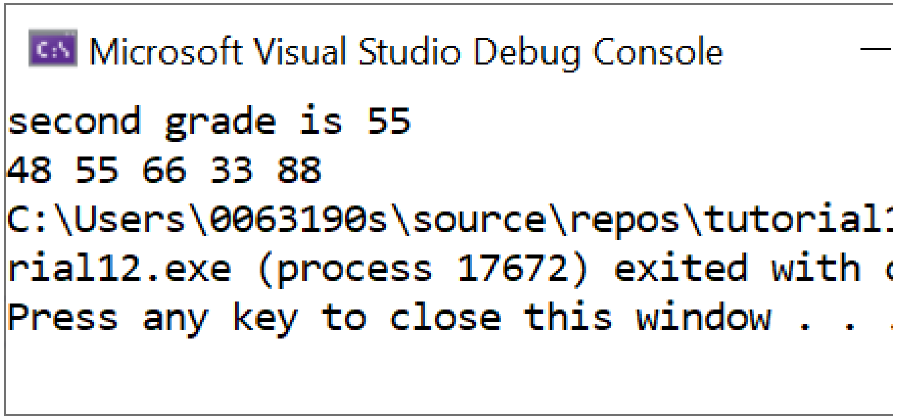
Remember

- The index of the array members always starts with 0, for example:
  - grades[0];

- For an array of length/size n (called myArray):
  - The indices range from 0 to n-1.
  - The elements range from myArray[0] to myArray[n-1].

Accessing array members

```
                  [0]  [1]  [2]  [3]  [4]
int grades[5] = { 44, 55, 66, 33, 88 };

grades[0] = 48; // easy to access/change any member of an array

printf("second grade is %d\n", grades[1]);

for (int i = 0;i < 5;i++)
{
  printf("%d ",grades[i]);
}
```



```
second grade is 55
48 55 66 33 88
C:\Users\0063190s\source\repos\tutorial1
rial12.exe (process 17672) exited with c
Press any key to close this window . . .
```

EXAMPLE PROBLEMS

Exercise Tracker App

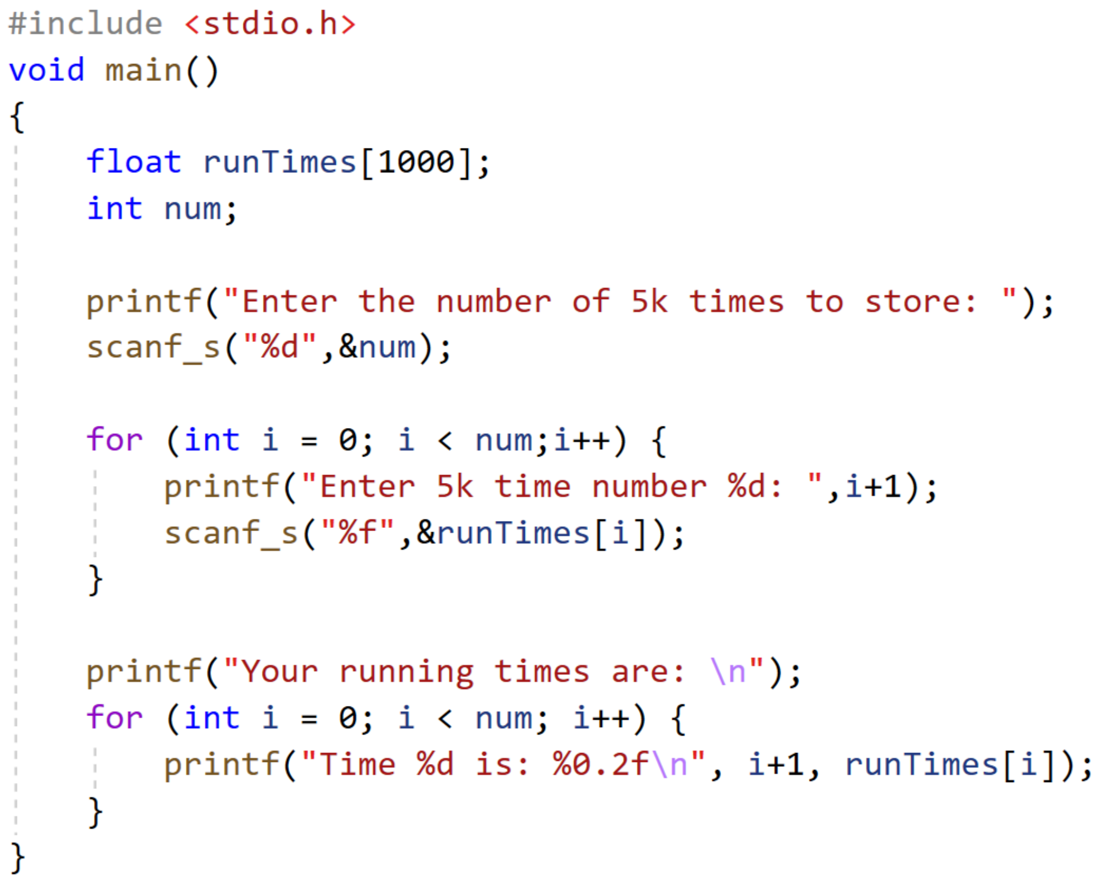
- You are designing an fitness app. The app allows the user to track their 5km running times.
- You must write a program that:
  - Reads in the number of 5km running times as input from the user.
  - Use loop to read in each 5km running time from the user.
  - Store these times in an array.
  - Print the running times out to the user so they can view them.

Exercise Tracker App

- Go to C program solution.

Exercise Tracker App

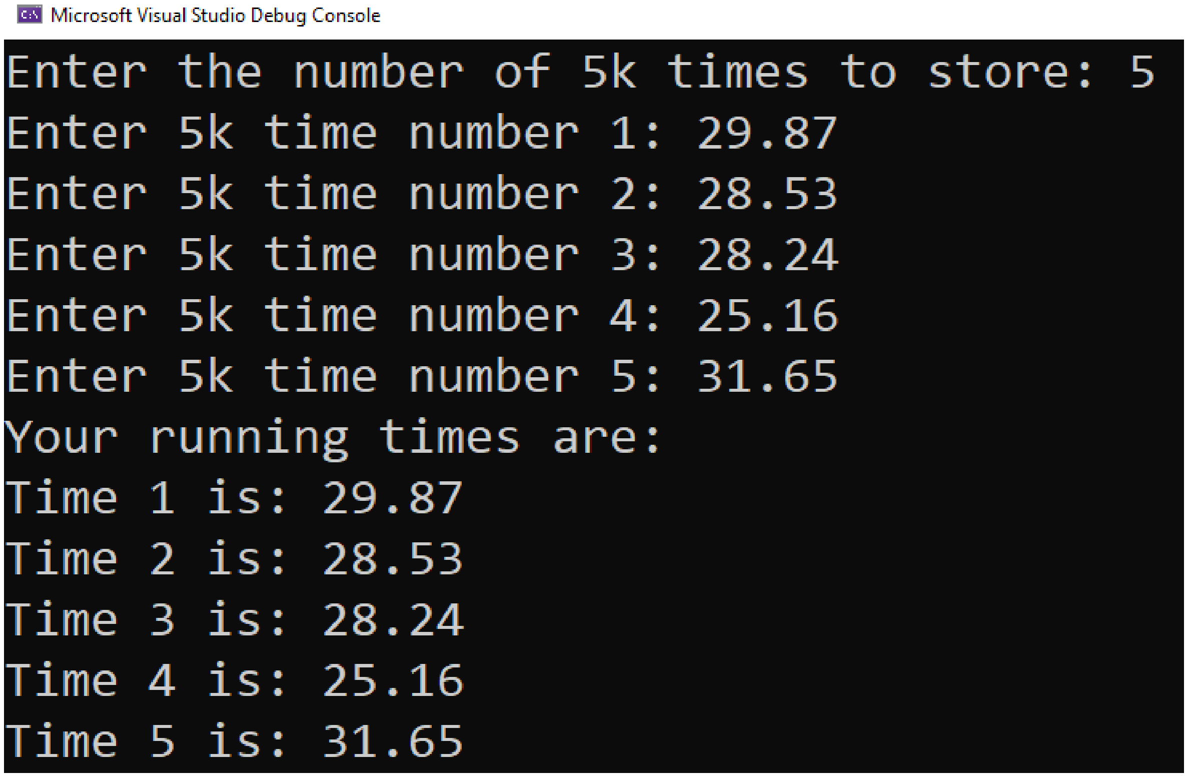
- The following code will work:



```
#include <stdio.h>
void main()
{
    float runTimes[1000];
    int num;

    printf("Enter the number of 5k times to store: ");
    scanf_s("%d",&num);

    for (int i = 0; i < num;i++) {
        printf("Enter 5k time number %d: ",i+1);
        scanf_s("%f",&runTimes[i]);
    }

    printf("Your running times are: \n");
    for (int i = 0; i < num; i++) {
        printf("Time %d is: %0.2f\n", i+1, runTimes[i]);
    }
}
```

Exercise Tracker App

- The code will produce the following output:



```
Enter the number of 5k times to store: 5
Enter 5k time number 1: 29.87
Enter 5k time number 2: 28.53
Enter 5k time number 3: 28.24
Enter 5k time number 4: 25.16
Enter 5k time number 5: 31.65
Your running times are:
Time 1 is: 29.87
Time 2 is: 28.53
Time 3 is: 28.24
Time 4 is: 25.16
Time 5 is: 31.65
```

Cinema Problem

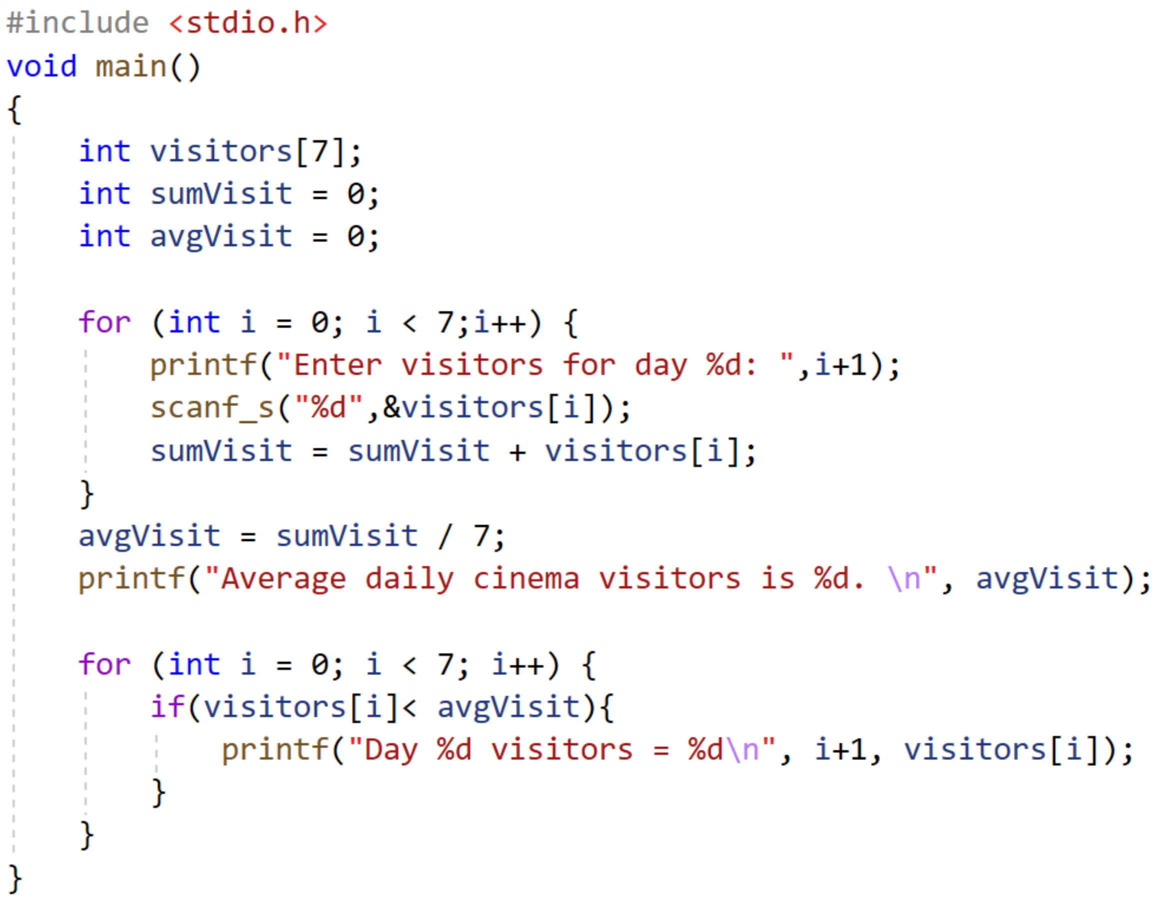
- You are designing software for a cinema. The cinema wants to record the daily visitors to the cinema over 7 days.
- You must write a program that:
  - Reads the number daily cinema goers as input from the user for 7 days.
  - Stores the cinema visitor numbers in an array.
  - Calculate and print the average number of daily cinema goers.
  - Prints out the daily visitor numbers that are below average.

Cinema Problem

- Go to C program solution.

Cinema Problem

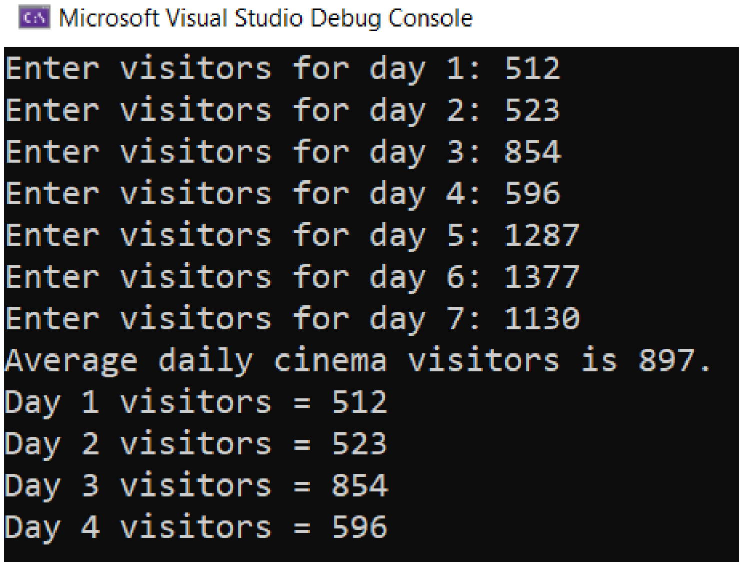
- The following code will work:



```c
#include <stdio.h>
void main()
{
    int visitors[7];
    int sumVisit = 0;
    int avgVisit = 0;

    for (int i = 0; i < 7;i++) {
        printf("Enter visitors for day %d: ",i+1);
        scanf_s("%d",&visitors[i]);
        sumVisit = sumVisit + visitors[i];
    }
    avgVisit = sumVisit / 7;
    printf("Average daily cinema visitors is %d. \n", avgVisit);

    for (int i = 0; i < 7; i++) {
        if(visitors[i]< avgVisit){
            printf("Day %d visitors = %d\n", i+1, visitors[i]);
        }
    }
}
```

Cinema Problem

- The code will produce the following output:
- Note: Be careful of rounding. We use integers when calculating the average here.



Microsoft Visual Studio Debug Console

```
Enter visitors for day 1: 512
Enter visitors for day 2: 523
Enter visitors for day 3: 854
Enter visitors for day 4: 596
Enter visitors for day 5: 1287
Enter visitors for day 6: 1377
Enter visitors for day 7: 1130
Average daily cinema visitors is 897.
Day 1 visitors = 512
Day 2 visitors = 523
Day 3 visitors = 854
Day 4 visitors = 596
```