

## CSE 130 Fall 2009

### Programming Assignment 3

Due Date: 11am, Monday Oct 19, 2009. Use `~cse130/submit` command from your sparky account. Do not forget to change to your working directory first. Include your name and ID number as a comment. Some points are reserved for this and for comments. Make sure that it runs on sparky.

**Objective:** Learn how to use one dimensional array.

**Important:** There are two parts to this program, but both parts are to be included in the same program. To get partial credit, submit whatever you have (before the deadline).

**Declaration:** First declare the array 'a' of type float and size 20. But before you do that, declare a constant `twenty = 20`, using constant declaration. You MUST USE `twenty` in declaration of the array a. You must also use constant 'twenty' in your program everywhere. Do not use number 20. You will lose some points. Main advantage of this is we can change size later, and we have to make only one change in our program right at the top. All the rest will be adjusted automatically.

#### Part I

**Initialization:** At the beginning, prompt user and ask for two positive integers m and n. Both should be less than 20. Do not check for errors. Next, initialize the array 'a' using the following expression, inside a loop. Use for or while loop. This loop runs from  $i = 0$  to  $(twenty-1)$  which is 19.

Calculate  $a[i] = (\text{float})(i*i + n)/(\text{float})(i+m)$ ;

After initialization, print the array such that 5 values are printed on each line. See the sample output format. An example will be posted soon.

**Printing the array:** For this you will need nested loops. Outer loop runs 4 times from  $k = 0$  to  $k = 3$ , for index/counter k for the outer loop. Inside this have one more loop that prints 5 values on a line. This loop runs from  $j = 0$  to  $j = 4$ , where j is the counter for the inner loop.

Next question is how to index array a using j and k? Here is an idea.  $a[k*5+j]$  will do the job. When  $k = 0$ , you will print  $a[0]$ , ...,  $a[4]$  on the 0th line. For the next line  $k = 1$ . So you will print  $a[5]$ , ...,  $a[9]$ , and so on. For the last line  $k = 3$ . So we will print from  $a[15]$ , ..., to  $a[19]$ .

While printing a floating point value of an array element, use "`%10.4f`" as a place holder. Here 10 is width of the field/number in digits, with 4 digits after the decimal points. Width 10 includes decimal point. THIS WILL ALIGN DECIMAL POINT FOR ALL LINES. You will need 4 lines for printing 5 values on each line. The initial space in "`%10.4f`", will skip a space before each number. Print `\t` (tab) at the beginning of each line. Print `\n` after 5 values have been printed on a line.

**Mean and Mean-Deviation:** Using a loop, add all values in array a, and calculate the mean or average. Print mean as shown. To obtain mean deviation, first get the positive difference  $|a[i] - \text{mean}|$  for the ith value in array a. Use an if statement for this. Depending on which one is larger, change the direction of subtraction. Add all such positive differences inside a loop that runs from  $i = 0$  to  $twenty-1$ . Divide the sum by twenty to obtain mean deviation. Print mean deviation as shown.

## Part-II

**Prefix sums:** For this part, calculate prefix sums of array *a* in another array 'ps' of size twenty (20). Calculate prefix sums and store them in array *ps* as follows.

$ps[0] = a[0]$ ,  $ps[1] = a[0] + a[1]$ ,  $ps[2] = a[0] + a[1] + a[2]$ , .... and so on.  
Finally,  $ps[19] = a[0] + a[1] + \dots + a[18] + a[19]$ .

Observe that instead of calculating array *ps* using the above definition, we could use another trick.

$ps[0] = a[0]$ ,  $ps[1] = ps[0] + a[1]$ ,  $ps[2] = ps[1] + a[2]$ , .... and so on.

In general  $ps[i] = ps[i-1] + a[i]$  for  $0 < i < \text{twenty}$  and  $ps[0] = a[0]$ .

Write one loop for calculating array *ps*. Write nested loops for for printing array 'ps'. Here again, print 5 values on each line. Follow the same idea described earlier.

Example session (on sparky):

Comments (**Do not print these**)

Sample output will be posted soon.

Type in integers *m* and *n*: 12 15

Array *a* after initialization is:

5 values on a line.

```
-----  
-----  
-----  
-----
```

Mean = .....

Mean and Mean-D

Mean deviation = .....

Array *ps* of prefix sums is:

```
-----  
-----  
-----  
-----
```

**Program structure is on the next page.**

## Program Structure:

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    const int twenty = 20;
```

```
    Declare array a and ps.
```

```
    Other declarations.
```

Part-I

Read m, n.

Initialize array 'a' using a loop.

Print array a with 5 values on each line.

Use \t at the beginning, and \n at the end of each line.

Calculate mean for array a, using one loop

Calculate mean deviation for array a, using another loop.

Print mean and mean deviation.

Part II

Obtain array 'ps' of prefix sums.

Print array 'ps' using the same technique for 'a'.

```
}
```