IPC 144 SLG – Midterm review

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Outline

1. Will it compile and execute?
2. Fill in the code
3. Code it

## Will it compile and execute?

Read the following snippets of code and verify if the program would compile.

If yes, what would be the output?

1. If it executes and asks for integers, assume that user will type ‘4 <enter> 9 <enter>’

(Note: <enter> is just a place holder for the action of pressing the ‘Enter’ key)

|  |  |
| --- | --- |
| **Code** | **Your Answer** |
| *#include <stdio.h>*  **struct** Employee {  **int** salary[50];  };  **int** main() {  **struct** Employee e;  **int** i, nElements = 10;  **for** (i = 5; i < 7; ++i) {  scanf(“%d”, &nElements);  }  **for** (i = 0; i < nElements; i++)  e.salary[i] = i + 1 \* 2;  **for** (i = 0; i < nElements; i++)  printf(“%d”, e.salary[i]);  printf(“\n”);  **return** 0;  } |  |

1. If it executes and prints to ask the user for a name, assume that the user will

type ‘John Appleseed <enter>’.

(Note: <enter> is just a place holder for the action of pressing the ‘Enter’ key)

|  |  |
| --- | --- |
| **Code** | **Your answer** |
| *#include <stdio.h>*  **struct** Student {  **int** studentNo;  **char** name[20];  };  **int** main() {  **struct** student name;  printf(*"Student's name: "*);  scanf(*"%s"*, name.name);  printf(*"You entered: "*, name.name);  **return** 0;  } | scanf(*"%s"*, &name.name);  printf("You entered: %s", name.name); |

1. If it executes and prints to ask the user for input, assume that the user will

type ‘15<space>5<enter>’.

(Note: <enter>and<space> is just a place holder for the action of pressing the ‘Enter’ or ‘Space’ key)

|  |  |
| --- | --- |
| **Code** | **Your answer** |
| *#include <stdio.h>*  **int** main()  {  **int** low, high;  printf(*"Enter two integers: "*);  scanf(*"%d %d"*, &low, &high);  **if** (low > high) {  **int** temp = high;  high = low;  low = temp;  }  **for** (; low != high; high--) {  low++;  printf(*"low = %d****\n****"*, low - 1);  }  **return** 0;  } |  |

## Fill in the code

1. Fill out the remainder pieces of the code **on page 4** to produce the following output

Name: **Mary**

Age: **23**

Grades: **90 85 91.5 89 96**

Name - Mike

Age – 21

Grades - 88.60 99.90 70.00 50.00 60.10

Name - Mary

Age - 23

Grades - 90.00 85.00 91.50 89.00 96.00

## Code it

1. Write a program **on page 5** that will prompt the user for positive whole numbers and print out the largest number they entered.

The program will begin with a brief introductory message saying what it will do. It will then ask the user to enter numbers and will keep doing this until they enter a value that is 0 or negative. It will then print out the largest value the user had entered (not including the 0 or negative value). If the user had entered no values, the program will display an error message saying that no numbers were entered.

Sample Runs (Your program must work to specification set above, this sample run is used only to clarify the way the program might look or feel)

|  |
| --- |
| **Sample run #1** |
| This program will tell the biggest number you entered  Enter a number (use 0 or negative value to terminate program): **5**  Enter a number (use 0 or negative value to terminate program): **90**  Enter a number (use 0 or negative value to terminate program): **7**  Enter a number (use 0 or negative value to terminate program): **10**  Enter a number (use 0 or negative value to terminate program): **92**  Enter a number (use 0 or negative value to terminate program): **42**  The biggest number you entered was 92 |

|  |
| --- |
| **Sample run #2** |
| This program will tell the biggest number you entered  Enter a number (use 0 or negative value to terminate program): **-1**  You did not enter any positive values |

### Fill in the code

### Question 1.

### **(Fill out your solution below)**

|  |
| --- |
| *#include <stdio.h>*    **struct** Student {  **char** name[30];  **int** age;  **float** grades[5];  };  **int** main(){  *// please notice that the second student is NULL, for now*  **struct** Student students[] = {  { *"Mike"*,  21,  { 88.6 f,  99.9 f,  70.0 f,  50.0 f,  60.1 f } },  NULL  };  // declare any necessary variables below  printf(*"Name: "*);  *// get the name of the second student from the user*  printf(*"Age: "*);  *// get the age of the student;*  printf(*"Grades: "*);  scanf(*"%f %f %f %f %f"*, &students[1].grades[0], &students[1].grades[1],  &students[1].grades[2], &students[1].grades[3], &students[1].grades[4]);  *// print the array*  **return** 0;  } |

### Code it

### Question 1.

### **(Write your solution below)**