

Problem 1 b (10 points) Write a complete C++ program that asks the user to enter their name and age. If the user is called Freddy and has an age that is either 7 or 17 the program should print `You won the special prize!` Otherwise the program should print `Sorry, please try again.`
Here is a sample to show how the program runs.

```
Enter the your name and age: Freddy 17
You won the special prize!
```

Answer:

```
#include <iostream>
using namespace std;

int main() {
    string name;
    int age;
    cout << "Enter the your name and age: ";
    cin >> name >> age;
    if (name == "Freddy" && (age == 7 || age == 17))
        cout << "You won the special prize!\n";
    else
        cout << "Sorry, please try again.\n";
    return 0;
}
```

Award partial credit for the following elements of a program:

- 1 points for overall structure: includes, main etc.
- 2 points for declaring the 2 variables.
- 1 point for prompt.
- 1 point for input of 2 variables.
- 4 points for the condition of the first if.
(1 for the `||`, 1 for the `&&` and 1 for the needed parentheses).
- 1 points for the else and printing

If a program follows a different (but reasonable) plan. Try to award partial credit for meeting similar goals in the program.

If you feel that a solution is too long and messy to grade properly (longer than about 25 lines if they were normally spaced out) judge a rough score for partial credit by the above milestones and then adjust the score down by the following guidelines.

- Very long and messy but looks probably correct --- max allowed score is 8/10.
- Very long and messy and looks partially correct --- max allowed score is 6/10.
- Very long and messy and looks probably badly wrong --- max allowed score is 4/10.

Problem 2 b (10 points) Consider the following C++ program. The program makes use of a function `first3digits` that returns the number formed by the first 3 digits of its input argument as its result. So for example, `first3digits(12345678)` would be 123.

Make sure to use your own 8-digit CUNY ID number as the number entered as input to the program. It would be a very bad idea to give answers based on another student's ID number!

```
int main() {
    int id, n, x = 10, y = 27, z = 16;

    cout << "Enter your 8-digit CUNY id number: ";
    cin >> id;    // assume that the user types YOUR OWN CUNY ID number

    cout << id << endl;                // line (a)
    n = first3digits(id);

    cout << n % x << endl;              // line (b)
    cout << n / x << endl;              // line (c)

    if ( (y < z) && ((x % y) < y) ) cout << "Yes\n"; // line (d)
    else cout << "No\n";

    y += 3; z /= 3;
    cout << y << z << y << endl;      // line (e)

    return 0;
}
```

(a) What is the output from the instruction beginning on line (a)?

Answer:

12345678

This answer is based on the ID number 12345678. Actual answers will be different.

(b) What is the output from the instruction beginning on line (b)?

Answer:

3

The answer will be the 3rd digit of the answer to (a).

(c) What is the output from the instruction beginning on line (c)?

Answer:

12

The answer will be the first two digits of the answer to (a).

(d) What is the output from the instruction beginning on line (d)?

Answer:

No

(e) What is the output from the instruction beginning on line (e)?

Answer:

30530

2 points per part. One point partial credit for answers that are almost correct except for a tiny error. (A tiny error might be extra or missing new lines or spaces.)

Check carefully that the 8 digit CUNY ID number is correct. If not email me what would be correct and what was used.

Note that the answers to parts b and c depend on the ID number.

Problem 3 b (10 points) Write a complete C++ program that repeatedly asks the user to enter a **size**. If the size is greater than or equal to 0, the program prints a square pattern of ***s** with that size. When a user enters a negative size, the program tells the user the total number of ***s** that have been printed and terminates.

Here is a sample to show how the program runs.

```
Enter a size or a negative number to stop: 2
**
**
Enter a size or a negative number to stop: 1
*
Enter a size or a negative number to stop: 2
**
**
Enter a size or a negative number to stop: -1
A total of 9 *s were printed.
```

Answer:

```
#include <iostream>
using namespace std;

int main() {

    int size = 0, total = 0;
    while (size >= 0) {
        for (int r = 1; r <= size; r++) {
            for (int c = 1; c <= size; c++) {
                cout << "*";
                total++;
            }
            cout << endl;
        }
        cout << "Enter a size or a negative number to stop: ";
        cin >> size;
    }
    cout << "A total of " << total << " *s were printed.\n";
    return 0;
}
```

Award partial credit for the following elements of a program:

- 1 point for overall structure: includes, main etc.
- 1 point for correct declaration and initialization of two variables.
- 1 points for a correct outer loop (likely while)
- 2 points for correct nested inner loops (likely for)
- 1 point for correctly printing *s
- 1 point for correctly printing lines
- 2 points for correctly incrementing total (there are many ways to place this)
- 1 point for correct output of total after all loops.

If a program follows a different (but reasonable) plan. Try to award partial credit for meeting similar goals in the program.

If you feel that a solution is too long and messy to grade properly (longer than about 25 lines if they were normally spaced out) judge a rough score for partial credit by the above milestones and then adjust the score down by the following guidelines.

Very long and messy but looks probably correct --- max allowed score is 8/10.
Very long and messy and looks partially correct --- max allowed score is 6/10.
Very long and messy and looks probably badly wrong --- max allowed score is 4/10.

Problem 4 b (10 points) The following program asks the user to enter a number n . It then prints a picture showing a triangle that points to the right that has $2n + 1$ rows and $n + 1$ columns. The odd numbered columns are made of `os` and the even ones are made of `*s`. For example, if $n = 3$ the program would print:

```
o
o*
o*o
o*o*
o*o
o*
o
```

Some pieces of code have been replaced by PART (a), PART (b), and so on. To answer the parts of this question you should supply the C++ code that was replaced. Each answer must fit on a single line.

```
int main() {
    int n;
    cout << "What is n? ";
    PART (a)
    for (int r = 1; PART (b); r++) {
        for (int c = 1; PART (c); c++) {
            if (PART (d)) cout << "*";
            else cout << "o";
        }
        PART (e)
    }
    for (int r = n ; PART (f); r--) {
        for (int c = 1; PART (g); c++) {
            if (PART (h)) cout << "*";
            else cout << "o";
        }
        PART (i)
    }

    return 0;
}
```

(a) Give a replacement for PART (a) to read the user's value of n

Answer: PART (a) is `cin >> n;`

(b) Give a replacement for PART (b) to loop over the upper rows of the picture:

Answer: PART (b) is `r <= n + 1`

(c) Give a replacement for PART (c) to loop over columns of the row:

Answer: PART (c) is `c <= r`

(d) Give a replacement for PART (d) to test whether to print a star

Answer: PART (d) is `c % 2 == 0`

(e) Give a replacement for PART (e) to finish each row

Answer: PART (e) is `cout << endl;`

(f) Give a replacement for PART (f) to loop over the lower rows of the picture:

Answer: PART (f) is `r >= 1`

(g) Give a replacement for PART (g) to loop over columns of the row:

Answer: PART (g) is `c <= r`

(h) Give a replacement for PART (h) to test whether to print a star

Answer: PART (h) is `c % 2 == 0`

(i) Give a replacement for PART (i) to finish each row

Answer: PART (i) is `cout << endl;`

1 bonus point for anyone who does anything on the problem.

Then 1 point per part.

This makes a total of 10.

Do not penalize students who write the whole line including the required missing part.