

Homework 2

Solution:

1. Please send solution to: zbaharav@cogswell.edu
2. You know the drill by now: Simply hit reply, and no sippze ddirectories etc.. Just ascii-files or Word documents (or equivalent)

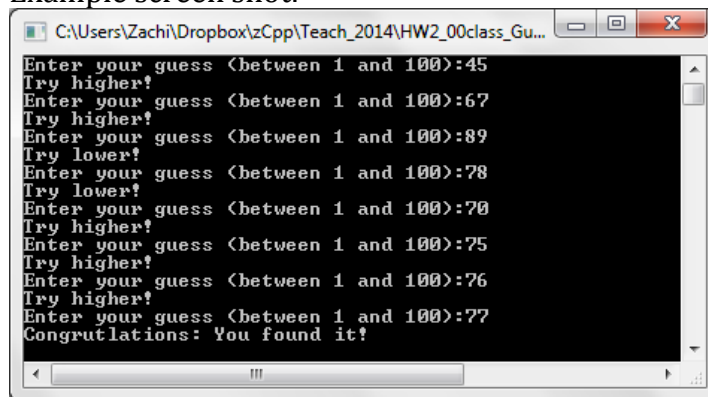
====

1. (taken almost 'verbatim' from Google educational material on C++)
Write a program that implements guessing game. Our program generates a random number between 0 and 100. The player must guess the secret number. The program give hints like "It's too high" or "It's too low", until the number is reached.

Work on this in three steps (just a guide. You still need to submit only one program):

- a. Figure out how to create a random number in a given range.
- b. Create a main function that processes one guess from the user, and provides hints.
- c. Add a loop to allow multiple guesses.

Example screen shot:



```
C:\Users\Zachi\Dropbox\zCpp\Teach_2014\HW2_00class_Gu...
Enter your guess <between 1 and 100>:45
Try higher!
Enter your guess <between 1 and 100>:67
Try higher!
Enter your guess <between 1 and 100>:89
Try lower!
Enter your guess <between 1 and 100>:78
Try lower!
Enter your guess <between 1 and 100>:70
Try higher!
Enter your guess <between 1 and 100>:75
Try higher!
Enter your guess <between 1 and 100>:76
Try higher!
Enter your guess <between 1 and 100>:77
Congrutlations: You found it!
```

====Solution=====

```
/*
  GuessTheNumber

  Computer chooses random number, and give you hints (up/down) until you guess it
  right.
*/

#include <iostream>
```

```

#include <time.h>
#include <cstdlib>

using namespace std;

int main(void)
{
    int number, guess = -1;

    srand (time(NULL));

    // Generate random number between 1 and 100
    number = rand() % 100 + 1;

    while (guess != number) {
        cout << "Enter your guess (between 1 and 100):";
        cin >> guess;

        if (guess<number)
            cout << "Try higher!\n";
        else if (guess>number)
            cout << "Try lower!\n";
    }

    cout << "Congrutlations: You found it!\n" ;

    return 0;
}

```

2. Write a program that reverse an integer input. Make sure to read the number as an integer, and print it as an integer. No need to transform to strings/characters/etc.

```

C:\Users\Zach\Dropbox\zCpp\Teach_2014\HW2_11home_reverse...
Please enter a number, integer, to be reversed:987321
Input number is: 987321
Reverse number is 123789
Program done!_

```

=====Solution=====

```

#include <iostream>
using namespace std;

int main()
{
    int num, mun=0;    // 'mun' is 'num' reversed
    int digit;

    cout << "Please enter a number, integer, to be reversed:";
    cin >> num;

```

```

cout << "\n\nInput number is: " << num << endl;

while(num)
{
    digit = num - (num/10)*10;
    mun = 10*mun + digit;

    num= num/10;
}

cout << "Reverse number is " << mun << endl ;
cout << "Program done!";

return 0;
}

```

3. PGM file format:

- a. Write two programs that read a PGM file, and write it back with the following modification:
 - i. Transform to Binary according to a user given Threshold.
 - ii. Creates an image half the size (Assume nice numbers for the input image)

Allow yourself plenty of room to make it easy !!

For example, if you are using the Baboon image, assume you know:

```

// Assume file header is like the following:
/*
P2
# comment line
512 512 <--- first number is cols, second is rows
255 <--- range of White
*/

```

So the program should do:

```

// Get the P2 line

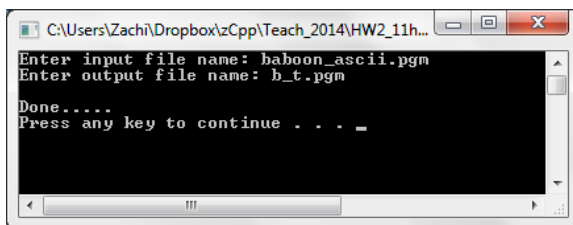
// Get the comment line

// Get the size of file

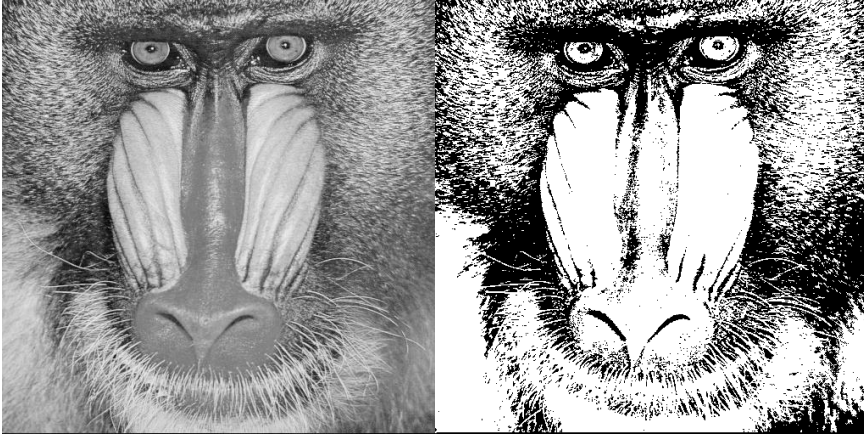
// Get the max gray level

// Get the values

```

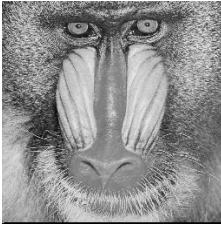


Threshold (with 127)
Left: Original. Right: Threshold.



```
C:\Users\Zachi\Dropbox\z\Cpp\Teach_2014\HW...
Enter input file name: baboon_ascii.pgm
Enter output file name: h_s.pgm
Done.....
Press any key to continue . . .
```

And Half the size:



====Solution=====

```
// Thresholding
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main()
{

    string fin_name, fout_name;

    // Get user input
    cout << "Enter input file name: ";
    cin >> fin_name ;

    cout << "Enter output file name: ";
    cin >> fout_name ;

    // Open files
    ifstream fin(fin_name);
```

```

if (!fin) // fin.fail()
{
    cout << "Failed to open file " << fin_name << " ;\nExiting...." ;
    exit(1);
}

ofstream fout(fout_name);
if (!fout) // fout.fail()
{
    cout << "Failed to open file " << fout_name << " ;\nExiting...." ;
    exit(1);
}

// Assume file header is like the following:
/*
P2
# comment line
512 512 <--- first number is cols, second is rows
255 <--- range of White
*/
string str;
// Get the P2 line
getline(fin, str);
fout << str << endl;

// Get the comment line
getline(fin, str);
fout << str << endl;

// Get the size of file
int cols, rows;
fin >> cols >> rows;
fout << cols << " " << rows << endl;

// Get the max gray level
int maxWhite;
fin >> maxWhite;
fout << maxWhite << endl;

int t = maxWhite/2; // threshold

for (int rr=1; rr<rows; ++rr)
{
    for (int cc=1; cc<rows; ++cc)
    {
        int val ;
        fin >> val;
        fout << ( (val > t)? maxWhite : 0) << " " ;
    }
    fout << endl;
}

cout << "\nDone.....\n" ;

fin.close();
fout.close();

```

```

        system("pause");
        return 0;
    }

    // Subsample

#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main()
{
    string fin_name, fout_name;

    // Get user input
    cout << "Enter input file name: ";
    cin >> fin_name ;

    cout << "Enter output file name: ";
    cin >> fout_name ;

    // Open files
    ifstream fin(fin_name);
    if (!fin) // fin.fail()
    {
        cout << "Failed to open file " << fin_name << " ;\nExiting...." ;
        exit(1);
    }

    ofstream fout(fout_name);
    if (!fout) // fout.fail()
    {
        cout << "Failed to open file " << fout_name << " ;\nExiting...." ;
        exit(1);
    }

    const int scale      = 2;

    // Assume file header is like the following:
    /*
    P2
    # comment line
    512 512  <--- first number is cols, second is rows
    255  <--- range of White
    */
    string str;
    // Get the P2 line
    getline(fin, str);
    fout << str << endl;

    // Get the comment line
    getline(fin, str);
    fout << str << endl;

```

```

// Get the size of file
int cols, rows;
fin >> cols >> rows;
fout << cols/scale << " " << rows/scale << endl;

// Get the max gray level
int maxWhite;
fin >> maxWhite;
fout << maxWhite << endl;

for (int rr=1; rr<rows; ++rr)
{
    for (int cc=1; cc<rows; ++cc)
    {
        int val ;
        fin >> val;

        if ( ((rr/scale)*scale == rr) && ((cc/scale)*scale ==
cc) )
            fout << val << " " ;
    }
    fout << endl;
}

cout << "\nDone.....\n" ;

fin.close();
fout.close();

system("pause");
return 0;
}

```

=== End of Homework 2 ===