1.

Input :

Enter your number for the pattern

5

Result:

1\*2\*3\*4\*5

11\*12\*13\*14\*15

21\*22\*23\*24\*25

16\*17\*18\*19\*20

6\*7\*8\*9\*10

2. Get n number of numbers from user. Remove duplicate value from that.

3.

Input :

Enter your number to print pattern

5

Result:

1\*

3\*2\*

4\*5\*6\*

10\*9\*8\*7\*

11\*12\*13\*14\*15\*

Program to print Full Pyramid using \*

4.

**Grading Program**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

Write a program that allows the user to enter the grade scored in a programming class (0-100).

If the user scored a 100 then notify the user that they got a perfect score.

★ Modify the program so that if the user scored a 90-100 it informs the user that they scored an A

★★ Modify the program so that it will notify the user of their letter grade

0-59 F 60-69 D 70-79 C 80-89 B 90-100 A

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Strings are your friends, until they betray you.**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

loops (for, while, do-while)

functions

strings & string functions

Write a program that asks for a user first name and last name separately.

The program must then store the users full name inside a single string and out put it to the string.

i.e.

Input:

John

Smith

Output:

John Smith

★ Modify the program so that it then replaces every a, e, i , o, u w/ the letter z.

i.e.

John Smith -> Jzhn Smzth

★★ Modify the Program so that it reverses the users name

i.e.

John Smith -> htimS nhoJ

5

**Cola Machine**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

Write a program that presents the user w/ a choice of your 5 favorite beverages (Coke, Water, Sprite, ... , Whatever).

Then allow the user to choose a beverage by entering a number 1-5.

Output which beverage they chose.

★ If you program uses if statements instead of a switch statement, modify it to use a switch statement.

If instead your program uses a switch statement, modify it to use if/else-if statements.

★★ Modify the program so that if the user enters a choice other than 1-5 then it will output "Error. choice was not valid, here is your money back."

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6**

**While( user == gullible )**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

loops (for, while, do-while)

Write a program that ccontinues to asks the user to enter any number other than 5 until the user enters the number 5.

Then tell the user "Hey! you weren't supposed to enter 5!" and exit the program.

★ Modify the program so that after 10 iterations if the user still hasn't entered 5 will tell the user "Wow, you're more patient then I am, you win." and exit.

★★ Modify the program so that it asks the user to enter any number other than the number equal to the number of times they've been asked to enter a number. (i.e on the first iteration "Please enter any number other than 0" and on the second iteration "Please enter any number other than 1"m etc. etc. The program must behave accordingly exiting when the user enters the number they were asked not to.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7**

**Pancake Glutton**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

loops (for, while, do-while)

arrays

Write a program that asks the user to enter the number of pancakes eaten for breakfast by 10 different people (Person 1, Person 2, ..., Person 10)

Once the data has been entered the program must analyze the data and output which person ate the most pancakes for breakfast.

★ Modify the program so that it also outputs which person ate the least number of pancakes for breakfast.

★★★★ Modify the program so that it outputs a list in order of number of pancakes eaten of all 10 people.

i.e.

Person 4: ate 10 pancakes

Person 3: ate 7 pancakes

Person 8: ate 4 pancakes

...

Person 5: ate 0 pancakes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8.

**Bracketing Search**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

loops (for, while, do-while)

psudo random numbers

Write a program that calculates a random number 1 through 100. The program then asks the user to guess the number.

If the user guesses too high or too low then the program should output "too high" or "too low" accordingly.

The program must let the user continue to guess until the user correctly guesses the number.

★ Modify the program to output how many guesses it took the user to correctly guess the right number.

★★ Modify the program so that instead of the user guessing a number the computer came up with, the computer guesses the number that the user has secretely decided. The user must tell the computer whether it guesed too high or too low.

★★★★ Modify the program so that no matter what number the user thinks of (1-100) the computer can guess it in 7 or less guesses.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**9,**

**Tic Tac Toe**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

loops (for, while, do-while)

arrays

Make a two player tic tac toe game.

★ Modify the program so that it will announce when a player has won the game (and which player won, x or o)

★★ Modify the program so that it is a one player game against the computer (with the computer making its moves randomly)

★★★★ Modify the program so that anytime the player is about to win (aka, they have 2 of 3 x's in a row, the computer will block w/ an o)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**10.**

**Dungeon Crawl**

Requires:

variables, data types, and numerical operators

basic input/output

logic (if statements, switch statements)

loops (for, while, do-while)

arrays

Make a program that outputs a simple grid based gameboard to the screen using either numbers or characters.

i.e.

. . . . . . . . . .

. G . . . . . . . .

. . . . . . T . . .

. . . . . . . . . .

. . . . T . . . . .

. . . . . . T . . .

. . . . . . . . . X

or

0 0 0 0 0 0 0 0 0 0

0 5 0 0 6 0 0 0 0 0

0 0 0 0 0 0 7 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 7 0 0 0 0 0 0

0 0 0 0 0 0 7 0 0 0

0 0 0 0 0 0 0 0 0 4

Allow the user (marked by G in the example) to move either up, down, left, or right each turn. If the player steps on a trap then they lose. If the make it to the treasure 'X' then they win.

★★ Add enemies that move randomly in any direction once per turn. (enemies just like traps cause the player to lose if touched)

HINT: Don't let the player move off the gameboard! You program will crash if they move off the top or bottom of the board!

(the same holds true for enemies)