

MIDTERM EXAM 2

EMPLID

CSCI 135 NAME: FIRST LAST

4. Write a function: `bool equals(char a[], int a_size, char b[], int b_size)`
that checks whether two `char` arrays are of equal length and have the same characters in the same order.

5. Write a function: `void bar_chart(int value_array[], int size)` that displays a bar chart of the values in `value_array`, using asterisks, like this:

```
*****
*****
*****
*****
*****
```

Your function must first check that all values in `value_array` are positive and no larger than 40. If a value falls outside of this range, no line should be printed for that value (not even blank one).

Variable and Constant Definitions

Type	Name	Initial value
int	cans_per_pack	6;
const double	CAN_VOLUME	0.335;

Mathematical Operations

#include <cmath>

pow(x, y)	Raising to a power x^y
sqrt(x)	Square root \sqrt{x}
log10(x)	Decimal log $\log_{10}(x)$
abs(x)	Absolute value $ x $
sin(x)	Sine, cosine, tangent of x (x in radians)
cos(x)	
tan(x)	

Selected Operators and Their Precedence

(See Appendix B for the complete list.)

[]	Array element access
++ -- !	Increment, decrement, Boolean not
* / %	Multiplication, division, remainder
+ -	Addition, subtraction
< <= > >=	Comparisons
= !=	Equal, not equal
&&	Boolean and
	Boolean or
=	Assignment

Loop Statements

Condition
while (balance < TARGET)
{
year++;
balance = balance * (1 + rate / 100);
}

Executed while condition is true

Initialization	Condition	Update
for (int i = 0; i < 10; i++)		
{		
cout << i << endl;		
}		

Loop body executed at least once

```
do
{
    cout << "Enter a positive integer: ";
    cin >> input;
}
while (input <= 0);
```

Conditional Statement

Condition
if (floor >= 13)
{
actual_floor = floor - 1;
}
else if (floor >= 0)
{
actual_floor = floor;
}
else
{
cout << "Floor negative" << endl;
}

Executed when condition is true

Second condition (optional)

Executed when all conditions are false (optional)

String Operations

```
#include <string>
string s = "Hello";
int n = s.length(); // 5
string t = s.substr(1, 3); // "ell"
string c = s.substr(2, 1); // "l"
char ch = s[2]; // 'l'
for (int i = 0; i < s.length(); i++)
{
    string c = s.substr(i, 1);
    or char ch = s[i];
    Process c or ch
}
```

Function Definitions

Return type	Parameter type and name
double	cube_volume(double side_length)
{	
double vol = side_length * side_length * side_length;	
return vol;	
}	Exits function and returns result.

Reference parameter

```
void deposit(double& balance, double amount)
{
    balance = balance + amount;
}
```

Modifies supplied argument

Arrays

Element type	Length
int	numbers[5];
int	squares[] = { 0, 1, 4, 9, 16 };
int	magic_square[4][4] =
{	
{ 16, 3, 2, 13 },	
{ 5, 10, 11, 8 },	
{ 9, 6, 7, 12 },	
{ 4, 15, 14, 1 }	
}	
for (int i = 0; i < size; i++)	
{	
Process numbers[i]	
}	