MIDTERM 2 Answer CS111 Summer 2020

NAME_____ CUNYID_____

Instructions:
1) Make sure your name and CUNY ID are filled in.

2) When asked to write a program, begin with the main portion of the program. In most cases I have written the beginning of the function for you and you just fill in the function body.

3) The only outside functions you can use are:
a) cout, cin, rand(), srand(), and time()
b) functions you write yourself
c) functions that appear anywhere on this test

SECTION 1 – Programming 5 Points each

Q1)

```
bool Contains(const char* str, const char c)
{
        int i = 0;
        while (str[i])
        {
                if (str[i] == c)
                        return(true);
                i++;
        }
        return(false);
}
```

Returns true if c is one of the following characters "~!@#$%^&*()_+"
Use a Contains helper function
IsSymbol('^')➔ true
IsSymbol('A')➔ false

```
bool IsSymbol(const char c)
{
        return(Contains("~!@#$%^&*()_+",c));
}
```

Q2)
Returns the N'th Upper Character
UpperNChar(0) ==> 'A'
UpperNChar(1) ==> 'B'

```
char UpperNChar(const int i)
{
```

```
        return('A' + i);
}
```

Q3)
convert character digit to number
CharToNum('0') ==> 0
CharToNum('9') ==> 9

```
int CharToNum(const char c)
{
        return(c - '0');
}
```

Q4)
returns true if c is alpha false otherwise
IsAlpha('&')➔ false
IsAlpha ('A')➔ true

```
bool IsAlpha(const char c)
{
        return((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'));
}
```

Q5)
returns the N'th lower case character
LowerNChar(0) ==> 'a'
LowerNChar (1) ==> 'b'

```
char LowerNChar (const int i)
{
        return('a' + i);
}
```

Q6)
returns true if a character is upper or lower case vowel false otherwise
A vowel is one of the letters A,E,I,O,U
IsVowel('A') ==> true
IsVowel('b') ==> false
IsVowel('i') ==> true
Use the Contains function

```
bool IsVowel(const char c)
{
        return(Contains("AEIOUaeiou",c));
}
```

Section 2 – Programming
Q1 10 Pts)
using a do while loop return the number of characters in a string, use as few lines as possible
StringLength("")==> 0

StringLength("ABC")==> 3

```
int StringLength(const char* str)
{
	int i=-1;
	do
	{
		i++;
	} while (str[i]);
	return(i);
}
```

Q2 10Pts)
returns the number of times num appears in nums
Count({1,2,3,4}, 2, 4) ==> 1
Count({1,1}, 2, 2) ==> 0
Count({1,1,3}, 1, 2) ==> 2

```
int Count(const int *nums, const int num, const int size)
{
	int count = 0;
	for (int i = 0; i < size; i++)
		if (nums[i] == num)
			count++;
	return(count);
}
```

Q3 a 3Pts)
convert digit to character
NumToChar(0) ==> '0'
NumToChar(9) ==> '9'

```
char NumToChar(const int i)
{
	return(i + '0');
}
```

Q3 b 3Pts)
returns Number Of Digits in a number
NumberOfDigits(1000) ==> 4
NumberOfDigits(10) ==> 2
NumberOfDigits(0) ==> 1

```
int NumberOfDigits(int i)
{
	int numOfDigits = 0;
	do
	{
		numOfDigits++;
		i /= 10;
```

```
        } while (i > 0);
}
```

Q3 c 6Pts)
using NumOfDigits and NumToChar convert return number as string
IntegerToString(11,str)==> "11"
IntegerToString(100,str)==> "100"

```
char *IntegerToString(int num, char* str)
{
        str[NumberOfDigits(num)] = 0;
        for (int i = NumberOfDigits(num) - 1; i >= 0; i--)
        {
                str[i] = NumToChar(num % 10);
                num /= 10;
        }
        return(str);
}
```

Q4a 5 Pts)
returns a string of first length characters in str
you can assume that str is always size of length + 1
Left("ABCDEF",3) ==> "ABC"
Left("ABCDEF",1) ==> "A"
Left("ABC",4) ==> "ABC"

```
int Min(int a, int b)
{
        return(a < b ? a : b);
}
char* Left(char* str, int length)
{
        str[Min(StringLength(str), length)] = 0;
        return(str);
}
```

Q4b 5Pts)
using a loop calculate the sum of the digits
SumOfDigits(123) ==> 6
SumOfDigits(221) ==> 5

```
int SumOfDigits(int n)
{
        int sum = 0;
        do
        {
                sum+=i%10;
                i /= 10;
```

```
        } while (i > 0);

        return(sum);
}
```

Q5 10 Pts)
output the numbers below using the one that occurs must frequently first
MostFrequently({3,4,1,1,1},5) ==> 1 1 1 3 4

```
void MostFrequently(const int* nums, int size)
{
        for (int i = size; i >= 1; i--)
        {
                for (int j = 0; j < size; j++)
                {
                        if (Count(nums, nums[j]) == i)
                                cout << (str[j]) << " ";
                }
        }
}
```

Q7 10 Pts)
using the IsVowel function return the number of vowels in the string

```
int NumberOfVowels(const char* str)
{
        int num = 0;
        int i = 0;
        while (str[i])
        {
                if (IsVowel(str[i]))
                        num++;
                i++;
        }
        return(num);
}
```

Q2a 3 Points)
convert character digit to number
CharToNum('0') ==> 0
CharToNum('9') ==> 9

```
int CharToNum(const char c)
{
        return(c - '0');
}
```

Q2b 7 Points)

convert string to integer use CharToNum
StringToInteger("100") ==> 100
StringToInteger("990") ==> 990
StringToInteger("090") ==> 90

```
int StringToInteger(const char* str)
{
	int i = 0;
	int answer = 0;
	while (str[i])
	{
		answer = (answer * 10) + CharToNum(str[i]);
		i++;
	}
	return(answer);
}
```

Q3a 3 Pts)
returns the number of times num appears in nums
Count({1,2,3,4}, 2, 4) ==> 1
Count({1,1}, 2, 2) ==> 0
Count({1,1,3}, 1, 2) ==> 2

```
int Count(const int *nums, const int num, const int size)
{
	int count = 0;
	for (int i = 0; i < size; i++)
		if (nums[i] == num)
			count++;
	return(count);
}
```

Q3b 3 Pts)
returns a random number between the two numbers
RandBetween(1, 2) ==> 1..2
RandBetween(10, 20) ==> 10..20

```
int RandBetween(int From, int To)
{
	srand(time(0));
	return(From + rand() % (To - From+1));
}
```

Q3c) 6 Pts
returns an array of random numbers between the From and To inclusive with no number appearing twice
using the RandBetween function above and the Count function
fill nums with random UNIQUE numbers between From and Two
RandBetween(nums, 5, 20, 30) ==> 20,30,29,28,27

RandBetween(nums, 1, 20, 30) ==> 21
RandBetween(nums, 2, 20, 30) ==> 21,21 (INVALID OUTPUT)

```
void RandBetween(int* nums, const int size, const int From, const int To)
{
	int count = 0;
	while (count < size)
	{
		int rand;
		do
		{
			rand = RandBetween(From, To);
		} while (Count(nums, rand, count)!=0);
		nums[count] = rand;
		count++;
	}
}
```

Q4 10 Pts)
returns the number of times that c appears in the string regardless of case
Count("AbcBB", 'b') ==> 3
Count("AbcBB", 'B') ==> 3
Count("AcdEfG", 'B') ==> 0

```
bool IsLower(const char c)
{
	return(c >= 'a' && c <= 'z');
}
char ToUpper(const char c)
{
	return(IsLower(c) ? 'A'+ c - 'a' : c);
}
int Count(const char *str, const char c)
{
	int i = 0,count=0;
	while (str[i])
	{
		if (ToUpper(str[i]) == ToUpper(c))
			count++;
		i++;
	}
	return(count);
}
```

Q5 10 Pts)
output the character that appears the most whether upper or lower case
if the more than one character appears the most times, output the first char

MostFrequently("AaABbBcd") ==> A
MostFrequently("AaABbBbcd") ==> B
MostFrequently("abcdefg") ==> a

```
char MostFrequently(const char* str)
{
	for (int i = StringLength(str); i >= 1; i--)
	{
		for (int j = 0; j < StringLength(str); j++)
		{
			if (Count(str, str[j]) == i)
				return(str[j]);
		}
	}
}
```