Row:	Seat:

Final Exam F22 V2
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

December 19, 2022

Exam Rules

- Show all your work. Your grade will be based on the work shown.

- The exam is closed book and closed notes with the exception of an 8 1/2" x 11" piece of paper filled with notes, programs, etc.

- When taking the exam, you may have with you pens and pencils, and your note sheet.

- You may not use a computer, calculator, tablet, phone, earbuds, or other electronic device.

- **Do not open this exam until instructed to do so.**

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures.

I understand that all cases of academic dishonesty will be reported to the Dean of Students and will result in sanctions.
Name:
EmpID:
Email:
Signature:

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C		
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

(Image from wikipedia commons)

1. (a) Fill in the code below to produce the output on the right:

```
weather = "Cloudy+Rain+Sunny+Snow+Windy"
```

i. `sunny = weather[`☐`]`
   `print(sunny)`

   Output:
   `Sunny`

   <mark>**Answer Key:**</mark>
   `sunny = weather[12:17]`
   `or`
   `sunny = weather[-16:-11]`

ii. `rain_snow =` ☐
   `for s in rain_snow:`
   `    print(`☐`)`

   Output:
   `RAIN`
   `SNOW`

   <mark>**Answer Key:**</mark>

```
1  rain_snow = weather.split('+')[1::2]
2  for s in rain_snow:
3      print(s.upper())
```

(b) Consider the following shell commands:

```
$ pwd
/usr/staff
$ ls
a.out p1_hello.py  p44_flower.py p60_binary.cpp
```

i. What is the output for:
   ```
   $ mkdir programs
   $ mv *.py programs
   $ rm a.out
   $ ls
   ```

   Output:
   ☐

   <mark>**Answer Key:**</mark>
   `p60_binary.cpp   programs`

ii. What is the output for:

Output:

```
$ cd programs
$ pwd
```

Answer Key:

/usr/staff/programs

iii. What is the output for:

```
$ cd ..
$ ls | grep cpp | wc -l
```

Output:

Answer Key:

1

2. (a) Select the color corresponding to the rgb values below:

    i. `rgb = (0, 0, 0)`
    ☐ black   ☐ red   ☐ cyan   ☐ gray   ☐ purple

    ii. `rgb = "#0000AB"`
    ☐ red   ☐ green   ☐ blue   ☐ black   ☐ white

    iii. `What is rgb values for purple?`
    ☐ 0, 0, 1   ☐ 0, 1, 1   ☐ 1, 0, 0   ☐ 1, 0, 1   ☐ 1, 1, 0

    iv. What is the binary number equivalent of decimal number 55?

    `Decimal 55 = Binary` [ ][ ][ ][ ][ ][ ]

    v. What is the Decimal number equivalent to Hexadecimal 3C?

    `Hexadecimal 3C = Decimal` [ ][ ]

Answer Key:

    i. `rgb = (0, 0, 0)`
    X black   ☐ red   ☐ cyan   ☐ gray   ☐ purple

    ii. `rgb = "#0000AB"`
    ☐ red   ☐ green   **X** blue   ☐ black   ☐ white

    iii. `What is rgb values for purple?`
    ☐ 0, 0, 1   ☐ 0, 1, 1   ☐ 1, 0, 0   **X** 1, 0, 1   ☐ 1, 1, 0

    iv. What is the binary number equivalent of decimal number 55?

```
        2  | 55
           +---
        2  | 27     1
           +---
        2  | 13     1
           +---
        2  |  6     1
           +---
        2  |  3     0
           +---
        2  |  1     1
           +---
              0     1
```
    `Decimal 50 = Binary` [1 | 1 | 0 | 1 | 1 | 1]

    v. What is the Decimal number equivalent to Hexadecimal 3C?
    Hexadecimal 3C = 3 * 16 + 12 = 3 * 16 + 12 = 60
    [6 | 0]

(b) Given the list `fruits` below, fill in the code to produce the Output on the right:

```
fruits = ['apple', 'banana', 'coconut', 'dragon fruit', 'elderberry']
```

i.
```
for j in range(            ):

        print(fruits[    ])
```

Output:
```
elderberry
coconut
apple
```

Answer Key:

```
1  fruits = ['apple', 'banana', 'coconut', 'dragon fruit', 'elderberry']
2
3  for j in range(4, -1, -2):
4      print(fruits[j])
5
6  You can also replace 4 by len(fruits)-1.
```

ii.
```
import numpy as np
import matplotlib.pyplot as plt
img = np.ones( (10,10,3) )

img[        ,        ] = 0
plt.imshow(img)
plt.show()
```

Output:

Answer Key:

```
import numpy as np
import matplotlib.pyplot as plt

img = np.ones( (10, 10, 3) )
img[5:, 5:] = 0
plt.imshow(img)
plt.show()
```
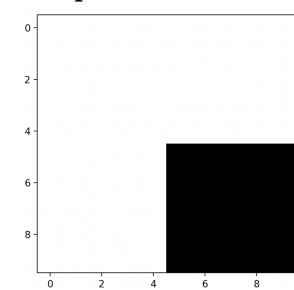
```
import numpy as np
import matplotlib.pyplot as plt
img = np.ones( (10,10,3) )

img[          ,          ] = 0
plt.imshow(img)
plt.show()
```

iii. **Answer Key:**

```
import numpy as np
import matplotlib.pyplot as plt

img = np.ones( (10, 10, 3) )
#img[0::2, 0:5] = 0 #work, 0 can be
    omitted
img[::2, :5] = 0
plt.imshow(img)
plt.show()
```

Output:
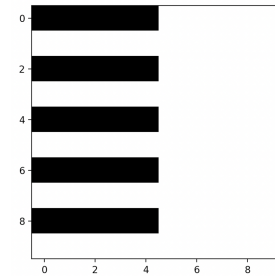
3. (a) What is the value (True/False):

         `in1 = True`
   i. `in2 = False`   ☐ True   ☐ False
         `out = not in1 and not in2`

   Answer Key:

   `out = False`

         `in1 = True`
         `in2 = False`
   ii. `in3 = False`   ☐ True   ☐ False
         `out = not (not in1 and not in2) or in3`

   Answer Key:

   `out = True`

         `in1 = True`
         `in2 = False`
   iii. `in3 = in1 or not in2`
         `out = not in1 or in2 or not in3`

                              ☐ True   ☐ False

   Answer Key:

   `out = False`

   iv.

         `in1 = True`
         `in2 = False`
         `in3 = True`
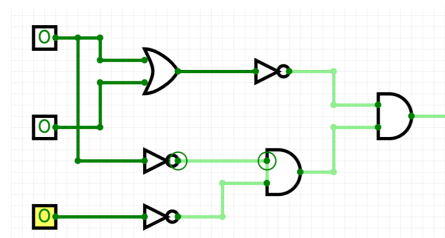
                              ☐ True   ☐ False

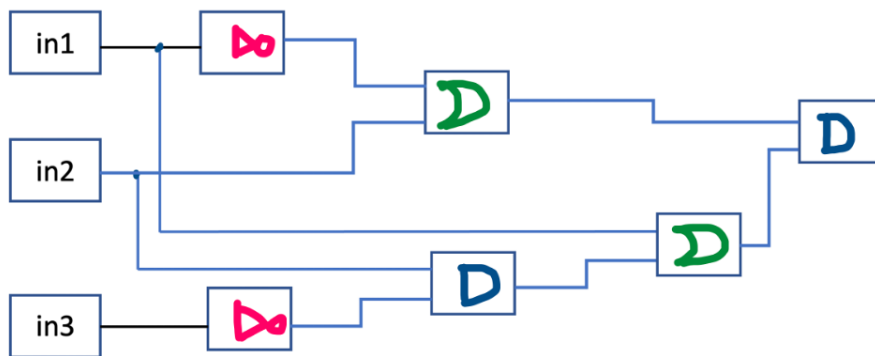   **Answer Key:**

   `out = False`

   (b) Draw a circuit that implements the logical expression:

   `(not in1 or in2) and (in1 or (in2 and not in3))`
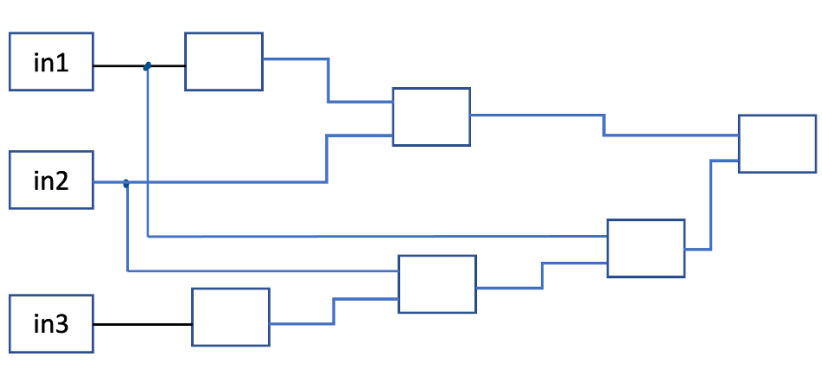
4. Consider the following functions:

```
def count(mylist, target):          def larger(s, t):
    num_occur = 0                       return s > t
    for num in mylist:
        if larger(num, target):     def main():
            num_occur += 1              brr = [3, 6, -5, 9, -2]
                                        print(count(brr, 3))
    return num_occur
```

(a) What are the formal parameters for `larger()`?

Answer Key: s, t

(b) What are the actual parameters for `count()`?

Answer Key: brr, 3

(c) How many calls are made to `larger()` after calling `main()`?

Answer Key: 5

(d) What is the output after calling `main()`?

Output:

Answer Key:

2

5. Design an algorithm that asks the user for the name of a text file containing a grid of numbers and loads it into a 2D array of integers(think like an image without the color channel) and a threshold. The program outputs the **sum** of all elements in the grid that are smaller than the threshold. For example, suppose the grid has values

```
[[1 2]
 [3 4]]
```

and the given threshold is 3. Then the sum is $1 + 2 = 3$.

Libraries:

Answer Key: numpy

Input:

Answer Key: the name of the text file, a number as a threshold

Output:

Answer Key: The total of all elements in the grid that are smaller than the threshold.

Design Pattern:
☐ Find Min   ☐ Find Max   ☐ Find All

Answer Key: ☐ Find Min   ☐ Find Max   **X** Find All

Principal Mechanisms (select all that apply):
☐ Single Loop   ☐ Nested Loop   ☐ Conditional (if/else) statement
☐ Indexing / Slicing   ☐ `split()`   ☐ `groupby()`

Answer Key:

☐ Single Loop   **X** Nested Loop   **X** Conditional (if/else) statement
XIndexing / Slicing   ☐ `split()`   ☐ `groupby()`

Process (as a concise and precise LIST OF STEPS / pseudocode):
(Assume libraries have already been imported.)

Answer Key:

(a) Ask the user for text file name

(b) Ask the user for a number as a threshold.

(c) Load data into grid.

(d) Set total to be zero.

(e) Use a nested loop to consider every element in grid, looping for rows in outer loop and columns in inner loop if the element is smaller than the threshold, add the element to total.

(f) Report total.

An implementation of the above code is as follows (This part is **optional** and will **not** be counted towards grading).

```
1  #suppose airtravel.csv has the following contents.
2  #"Month", "1958", "1959", "1960"
3  #"JAN", 340, 360, 417
4  #"FEB", 318, 342, 391
5  #"MAR", 362, 406, 419
6  #"APR", 348, 396, 461
7  #"MAY", 363, 420, 472
8  #"JUN", 435, 472, 535
9  #"JUL", 491, 548, 622
10 #"AUG", 505, 559, 606
11 #"SEP", 404, 463, 508
12 #"OCT", 359, 407, 461
13 #"NOV", 310, 362, 390
```

```
14   #"DEC", 337, 405, 432
15
16   import numpy as np
17
18   in_file = input("Enter input file name: ")
19   threshold = int(input("Enter a threshold: "))
20   grid = np.loadtxt(in_file, skiprows=1, delimiter=',', usecols=range(1,4))
21   #skip the first row, which is column head
22   #skip the first column, since it is row head
23   #print(grid)
24
25   numRows = grid.shape[0]
26   numCols = grid.shape[1]
27
28   total = 0
29   for i in range(numRows):
30       for j in range(numCols):
31           if grid[i,j] < threshold:
32               total += grid[i,j]
33
34   print(total)
```

6. Consider the `violations.csv` dataset that reports violations issued by Business Integrity Commission for companies operating in the trade waste industry. A snapshot given in the image below:

VIOLATION N	VIOLATION ACCOUNT STATE	FINE AMOUNT	NUMBER OF COUNTS	DESCRIPTION OF RULE
TWC-219653	NJ	500	1	Removed collected or disposed (
TWC-218679	NJ	1000	1	Failed to timely notify Commiss
TWC-211037	NY	2500	1	Removed collected or disposed (
TWC-221854	NY		1	Removed collected or disposed (
TWC-218495	NY	0	1	Failed to separate recyclable ma

Assume we write `import pandas as pd` already. Fill in the Python program below:

```
#Read input data into data frame:

df =
```

```
#Print the average value in column 'NUMBER OF COUNTS'.
```

```
#Groups the data by 'VIOLATION ACCOUNT STATE' to extract data in NY.

ny =
```

```
#Print the maximum of FINE AMOUNT in NY.
```

```
#Print the most common (aka top) FIVE rules violated.
#Hint: look at 'DESCRIPTION OF RULE' and value_counts method.
```

Answer Key:

```
1  #To test, download https://data.cityofnewyork.us/Business/BIC-Issued-Violations
       /upii-frjc,
2  #shorten the file name as violations.csv.
3  import pandas as pd
4
5  df = pd.read_csv("violations.csv")
6  print(df["NUMBER OF COUNTS"].mean())
7  ny = df.groupby("VIOLATION ACCOUNT STATE").get_group("NY")
8  print(ny['FINE AMOUNT'].max())
9  print(df["DESCRIPTION OF RULE"].value_counts()[:5])
```

7. Complete the following code in Python.

Define `sameFreq` function, for strings s1 and s2, char ch, see whether s1 and s2 have the same number of occurrences of ch. For example, the return of `sameFreq('abc', 'acd', 'a')` is true, but the return of `sameFreq('abc', 'acd', 'b')` is false.

Define `allSameFreq` function, for strings s1, s2, and s3, check whether s1 and s2 have the same number of occurrences of each letter in s3. For example, allSameFreq('abcd', 'bcae', 'abc') returns true, but allSameFreq('abcd', 'bcae', 'abd') returns false.

Hints: for the first letter in s3 that does not have the same number of occurrences in s1 and s2, can you stop and know what `allSameFreq` function should return immediately? What if after testing every letter in s3, and each one has the same number of occurrences in s1 and s2?

Answer Key:

```python
def sameFreq(s1, s2, ch):
    return s1.count(ch) == s2.count(ch)

def allSameFreq(s1, s2, s3):
    for ch in s3:
        if not sameFreq(s1, s2, ch):
            return False

    return True

def main():
    print(allSameFreq('abcd', 'bcae', 'abc'))
    print(allSameFreq('abcd', 'bcae', 'abd'))

if __name__ == '__main__':
    main()
```

8. (a) What does the MIPS program below print:

Output:

[blank box]

<mark>**Answer Key:**</mark>

```
egikm
```

(b) Modify the program to print out string "975". Shade in the box for each line that needs to be changed and rewrite the instruction below. Warning: you need to modify from the above code. Need to use j and beq commands.

☐ ADDI $sp, $sp, -6 # Set up stack

☐ ADDI $t0, $zero, 101 # Set $t0 at 102 ('e')

☐ ADDI $s2, $zero, 5 # Use to test when you reach 5

☐ SETUP: SB $t0, 0($sp) # Next letter in $t0

☐ ADDI $sp, $sp, 1 # Increment the stack

☐ ADDI $s2, $s2, -1 # Decrement the counter by 1

☐ ADDI $t0, $t0, 2 # Increase the letter by 2

☐ BEQ $s2, $zero, DONE # Jump to DONE if s2 == 0

☐ J SETUP # Else, jump back to SETUP

☐ DONE: ADDI $t0, $zero, 0 # Null (0) to terminate string

☐ SB $t0, 0($sp) # Add null to stack

☐ ADDI $sp, $sp, -5 # Set up stack to print

☐ ADDI $v0, $zero, 4   # 4 is for print string

☐ ADDI $a0, $sp, 0   # Set $a0 to stack pointer

☐ syscall   # Print to the log

Answer Key:

```
1  ADDI $sp, $sp, -4         # Set up stack
2  ADDI $t0, $zero, 57         # Set $t0 at 57 ('9')
3  ADDI $s2, $zero, 3         # Use to test when you reach 3
4  SETUP: SB $t0, 0($sp)     # Next letter in $t0
5  ADDI $sp, $sp, 1          # Increment the stack
6  ADDI $s2, $s2, -1         # Decrement the counter by 1
7  ADDI $t0, $t0, -2         # Decrease the letter by 3
8  BEQ $s2, $zero, DONE      # Jump to DONE if s2 == 0
9  J SETUP                   # Else, jump back to SETUP
10 DONE: ADDI $t0, $zero, 0  # Null (0) to terminate string
11 SB $t0, 0($sp)            # Add null to stack
12 ADDI $sp, $sp, -3         # Set up stack to print
13 ADDI $v0, $zero, 4        # 4 is for print string
14 ADDI $a0, $sp, 0          # Set $a0 to stack pointer
15 syscall                   # Print to the log
```

9. Fill in the C++ programs below to produce the Output on the right.

```cpp
#include <iostream>
using namespace std;
int main()
{

    for(int i = 2; i <= [        ]  ;  [        ]  ){

        cout << i/2  << endl;
    }
    return 0;
}
```

Answer Key:

(a)
```cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 2; i <= 14 ; i+=4 )
7      //Warning: do not add ; right after for-head,
8      //or the loop body is empty.
9      //That is,
10     //the following writing is WRONG
11     //for (int i = 2; i <= 14; i+=4) ;
12     //We say, each statement in C++ ends with ;
13     //we do not say, each line in C++ ends with ;
14     {   //this pair of curly braces can be omitted,
15         //since the loop body has only statement
16         cout << i / 2 << endl;
17     }
18
19     return 0;
20 }
```

Output:

1
3
5
7

(b)

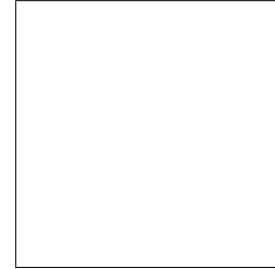```cpp
#include <iostream>
using namespace std;

int main()
{
    int size = 5;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < i; j++)
            cout << " ";
        for (int j = 0; j < size - i; j++)
            cout << "*";
        cout << endl;
    }

    return 0;
}
```

**Output:**

**Answer Key:**

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

(c)

```cpp
#include <iostream>
using namespace std;
int main(){
    int m = 2;
    int n = 6;

    while ( m + n <= [        ] ) {
        cout << m << " " << n << endl;

        [              ] //update m

        n--;
    }
    return 0;
}
```

Output:

```
2 6
7 5
12 4
17 3
```

Answer Key:

```
m + n <= 20
```

```
n += 5; or
n = n+5;
```

A complete C++ code is as follows.

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

int main()
{
    int m = 2;
    int n = 6;
    while (m + n <= 20)
    {
        cout << m << " " << n << endl;
        m += 5;
        n --;
    }

    return 0;
}
```

10. (a) Translate the following python program into a **complete C++ program**:

```python
num = -1
while num < 0 or num > 100:
    num = int(input("Enter an integer in [0, 100]: "))

print("num =", num)
```

//include library and namespace

//main function signature

{
  //initialization

  //loop line

  //loop body
  {

  }
  //print num

  //return

```
    }
```

Answer Key:

```cpp
1   #include <iostream>
2   using namespace std;
3
4   int main()
5   {
6       int num = -1;
7       while (num < 0 || num > 100)
8       {
9           cout << "Enter an integer in [0, 100]: ";
10          cin >> num;
11      }
12
13      cout << "num = " << num << endl;
14
15      return 0;
16  }
```

(b) Write a C++ code. Declare variables for lb and kilograms (kg). Declare variable for choice. If choice is 1, then enter number of lbs, and convert it to kilograms (kgs) and print the result out. Otherwise, enter number of kgs, and convert it to lbs and print the result out.

1 lb = 0.45 kg

1 kg = 1 / 0.45 lbs

Some sample input/output is as follows.

```
Enter a choice: 1
Enter number of lbs: 2
2 lbs = 0.9 kgs

Enter a choice: 2
Enter number of kgs: 3
3 kgs = 6.66667 lbs
```

Just finish the code in main function. No need to write include library and main function signature and return statement.

```
//declare variables lbs and kgs (for kilograms).
```

```
//declare and obtain input for variable choice.
```

```
//Write if-statement when choice is 1,
//input lbs, convert to kgs (kilograms), and output result.
```

```
//Write else-statement, input kgs, convert to lbs, and output result.
```

<mark>**Answer Key:**</mark>

```cpp
1   #include <iostream>
2   using namespace std;
3
4   int main()
5   {
6       //declare variables for lbs and kgs.
7       double lbs;
8       double kgs;
9
10      //declare and input for variable choice
11      int choice;
12      cout << "Enter a choice: ";
13      cin >> choice;
14
15      //when choice is 1
16      if (choice == 1)
17      {
18          cout << "Enter number of lbs: ";
19          cin >> lbs;
20          kgs = lbs * 0.45;
21          cout << lbs << " lbs = " << kgs << " kgs" << endl;
22      }
23      else //when choice is not 1
24      {
25          cout << "Enter number of kgs: ";
26          cin >> kgs;
27          lbs = kgs / 0.45;
28          cout << kgs << " kgs = " << lbs << " lbs" << endl;
29
30      }
31      return 0;
32  }
```

SCRATCH PAPER

SCRATCH PAPER