

Make Up Set

Total Points: 30

Choose 3 of the 6 problems below for full credit. The starred problem must be implemented. See 1436 Grading Guide Sheet for additional grading/submission information. Partial credit will be given. (10 points each)

1. A clothing store is having a sale. They need to make new sale tags for everything in the store. They have a list of every item in the store along with the item's original and new marked-down price. Write a C++ program that will read the item from the keyboard and generate each item's sale tag. The sale tag should show the name of the item as well as the item's markdown given as a percentage of its original price. Markdowns will always be at least 1%. Item names will contain only letters and possibly spaces, and markdowns should be truncated (use a cast) if necessary (so 10.2% and 10.7% would both be printed as just 10%). Some needed formulas:

$$\text{discount} = \text{marked price} - \text{selling price}$$

Expressing the discount as a percentage of the marked price:

$$\text{discount\%} = \frac{\text{\$discount}}{\text{\$marked price}} \times 100\%$$

Assume valid input. Refer to the sample output below.

Sample Run

```
Enter the item: Gloves
Enter old price: 9.99
Enter new price: 6.49
```

```
Gloves! Now 35% off!
```

Name the program: DiscountPriceXX.cpp, where XX are your initials.

2. In a momentary loss of computing ability, Professor C. needs you to translate the following tricky formulas. In his madness, he chose to name these formulas after his 4 cats: Java, CPlus, CSharp, and Python.

$$\text{Java} = \frac{A}{4} + 7B$$

$$CPlus = \frac{A+B^2}{A+\frac{1}{B}} + \frac{A}{B}$$

$$CSharp = \frac{1}{\frac{1}{A} - \frac{1}{B}}$$

$$Python = \frac{4}{\frac{A}{B}} \left[\frac{1 + \frac{5}{C+D}}{A} \right]^{\frac{1}{2}} - \frac{A}{C+D}$$

Write a C++ program to calculate the formula results. Let the user input four **decimal numbers (double)** for A, B, C and D (in that order) from the keyboard. Output and label the value of the expressions Java, CPlus, CSharp, and Python (in that order). An error tolerance of ± 0.01 will be accepted. Output labeled answers to two decimal places. Output should be user friendly.

Name the program: TrickyFormulasXX.cpp, where XX are your initials.

3. The chili peppers you eat can be measured in terms of their "hotness " or "heat". The Scoville scale of measurement is used to determine this value. Write a C++ program that prompts the user for a hotness value and prints the corresponding pepper found in the chart below.

Pepper Name	Scoville Heat Units
Carolina Reaper Pepper	855,000 – 2,200,000
Red Savina Habanero Pepper	350,000 – 580,000
Habanero Chili Pepper	100,000 – 350,000
Piri Piri Pepper	50,000 – 100,000
Cayenne Pepper	30,000 – 50,000
Peter Pepper	10,000 – 30,000
Jalapeno Pepper	3,500 – 10,000
Poblano Pepper	1,000 – 3,500
Banana Pepper	100 – 1,000
Bell Pepper	0 – 99

Note the gap in values. Use constants and error check. Output should be user friendly.

Name the program: HotPeppersXX.cpp, where XX are your initials.

4. If a number is the only number between a prime number and a square number it is beprisque. Write a C++ program that that asks to the user to check to see if the number entered is beprisque or not. Check for numbers between 1 and 10000 only. The first 10 beprisque numbers are: 1, 2, 3, 8,

10, 24, 48, 80, 82, 168. The values must not be hardcoded. A user defined function must be used. Refer to the sample output below.

Sample Runs:

Enter a number: 24
24 is a beprisque number.

Enter a number: 300
300 is not a beprisque number.

Name the program: BeprisqueXX.cpp, where XX are your initials.

5. Write a C++ program to determine if the player has been dealt a flush, straight, straight flush, royal flush or nothing (high card). A poker hand of five playing cards drawn from a standard 52-card deck has a value, based on the modified table below. Each card has a rank (2–9, 10, Jack, Queen, King, or Ace) and a suit (♣, ♦, ♥, or ♠). A card is represented as a 2-character string with a rank of (A, 2–9, T, J, Q, or K) and a suit of (C, D, H, or S). Thus, "AS" represents the Ace of Spades, while "TC" represents the Ten of Clubs.

Power	Poker Hand Value	Description	Examples
10	ROYAL FLUSH	The 10, Jack, Queen, King and Ace of the same suit. This hand is the strongest of all poker hands.	A♦ K♦ Q♦ J♦ T♦
9	STRAIGHT FLUSH	Five cards in sequence. Aces can play low or high (default). All cards belong to the same suit. Note that Ace may not "wrap around" and play both high and low.	Q♣ J♣ T♣ 9♣ 8♣ or 5♦ 4♦ 3♦ 2♦ A♦
6	FLUSH	All five cards are of the same suit, but not in sequence	Q♦ 9♦ 7♦ 4♦ 3♦
5	STRAIGHT	Five cards in sequence using at least two different suits. Aces can play low or high. Note that Ace may not "wrap around" and play both high and low.	A♣ K♣ Q♦ J♠ T♠ or 5♠ 4♦ 3♦ 2♠ A♥
1	HIGH CARD	Five cards not meeting any of the above requirements.	K♥ J♥ 8♣ 7♦ 4♠

Hands will be input from the keyboard with five card representations, separated by a single space between each card (convert to uppercase). If the user enters an illegal card or tries to enter the same card twice, the program will ignore the card, issue an error, and then request another card. The program should include at least two user defined functions to calculate the hand strength. The program should

also include a loop that lets the user repeat the program until the user says he/she is done. Check validation. Use one dimensional arrays. Refer to the sample output below.

Sample Runs (2):

```
Deal five cards: 8C JC 9H TS QD
This is a Straight.
```

```
Run again(Y/N): Y
```

```
Deal five cards: 2D 5S 3C 4C 9H
This is nothing- High Card
```

```
Run again(Y/N): n
```

Name the program: FlushStraightsXX.cpp, where XX are your initials.

6 (**). Write a C++ program that fills the right to left diagonal of a square matrix with zeros, the lower-right triangle with -1s, and the upper left triangle with +1s. Let the user enter the size of the matrix up to size 21. Use a constant and error check. Output the formatted square matrix with appropriate values. Refer to the sample output below.

Sample Run:

```
*** Start of Matrix ***
```

```
Enter the size of square (<= 21): 5
```

```
1 1 1 1 0
1 1 1 0 -1
1 1 0 -1 -1
1 0 -1 -1 -1
0 -1 -1 -1 -1
```

```
*** End of Matrix ***
```

Name the program: FillMatrixXX.cpp, where XX are your initials.

Extra Credit: Implement the problems below for extra credit. See 1436 Grading Guide Sheet for additional grading/submission information. Partial credit will be given.

1. Write a C++ program, when given a square grid of non-negative integers, you must find the greatest product of any four adjacent numbers in any direction (up, down, diagonally, left, right). All numbers in the grid will be less than 100. Input from a data file consisting of a square grid of integers. Numbers on the same line will be separated by a single space. The program should print out the greatest product of any four adjacent numbers in the grid using the format shown below. The solution is highlighted below for clarity. Let the user input the file name from the keyboard. Refer to the sample output below.

Sample File

```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

Sample Run:

Enter the file name: grid.txt

Greatest product: 70600674

Name the program: GridProductXX.cpp, where XX are your initials.

2. Write a C++ program to manipulate "big letters". To be snarky, write a C++ program that displays letters in a really big font:

```

      A   BBBB   CCCC DDDD   EEEEE FFFFF   GGG   H   H   IIIII JJJJJ K   K   L
    A A  B   B C       D   D E       F       G       H   H   I       J   K   K   L
  AAAAA BBBB  C       D   D EEE   FFF   G   GG HHHHH   I       J   KKK   L
  A   A B   B C       D   D E       F       G   G H   H   I   J   J   K   K   L
  A   A BBBB   CCCC DDDD   EEEEE F       GGG   H   H   IIIII   JJ   K   K   LLLLL

```

Input a one-word string to be displayed in the big font with 26 letters. Convert to uppercase on input. Output each of the letters using the format displayed above. Note that each letter is 5 x 5 and are separated by a space of width 1. Use the C++ string class. Use a function in the calculation. Refer to the sample output below.

Note: All 26 letters are not shown above due to space.

Sample Run:

Enter a name: Horner

```

H   H   OOO   RRRR   N   N   EEEEE   RRRR
H   H O   O R   R NN   N E       R   R
HHHHH O   O RRRR   N N N   EEE   RRRR
H   H O   O R   R N   NN E       R   R
H   H   OOO   R   R N   N   EEEEE   R   R

```

Name the program: BigLettersXX.cpp, where XX are your initials.