



1. Exponentiation
2. Print and Sum Multiples
3. Fahrenheit / Celsius Table
4. Count Characters
5. Quit

You must check that a valid menu selection has been given. If it is invalid, an error message should be displayed and the menu/input prompt repeated until valid data is given. The program should continue until the user selects the quit option (i.e., the only way the program terminates is when the user selects the quit option).

### **Calculations and Output**

All floating point numbers must be displayed with 3 decimal places of precision.

Exponentiation: Requires input of two integers. You are NOT allowed to use the predefined function `pow`. The first integer input is the base. The second is the exponent, which you may assume is  $\geq 0$ . You are to use a loop to perform an exponentiation calculation. Display the result when done.

Multiples: Requires the input of three integers. The first integer indicates what type multiple is desired, e.g., multiples of 3 or multiples of 5. The second and third integers are the beginning and ending value (inclusive). For example, if the integers 4, 20 and 50 are input, this means the program must display all multiples of 4 between 20 and 50 (inclusive) AND sum those values as they are printed. The sum must be displayed once after the completion of the loop that displays the multiples.

Fahrenheit / Celsius: Requires the input of two integers. These integers are the beginning and ending fahrenheit values (inclusive) of the temperatures to be displayed. The formula to convert Fahrenheit to Celsius is: `Celsius = (Fahrenheit - 32) * 5.0/9.0;`

Count Characters: Requires input of a sentinel (string) value. This option should repeatedly ask the user for a single word, using the sentinel value to terminate the loop/program. As each word is input by the user, if it is not the sentinel value, accumulate the sum of the lengths of the words input. The predefined function and syntax for determining string length is:

```
strLength = string1.length();
```

where `strLength` is an integer variable, and `string1` is a string variable. Be sure to include the string library: `#include <string>`

**Again, you MUST name your source file 'pa6.cc' and you MUST store it in your class directory (cs150x).**

See next page for Sample Run !!!

**Sample Run:**

1. Exponentiation
2. Print and Sum Multiples
3. Fahrenheit / Celsius Table
4. Count Characters
5. Quit

Choice: 1

Enter the base integer: 4  
Enter the exponent integer: 5

4 raised to the power 5 = 1024

1. Exponentiation
2. Print and Sum Multiples
3. Fahrenheit / Celsius Table
4. Count Characters
5. Quit

Choice: 2

Enter the multiple(integer) to use: 6  
Enter beginning integer: 20  
Enter ending integer: 100

24 30 36 42 48 54 60 66 72 78  
84 90 96

The sum of these multiples of 6 is: 780

1. Exponentiation
2. Print and Sum Multiples
3. Fahrenheit / Celsius Table
4. Count Characters
5. Quit

Choice: 3

Enter beginning Fahrenheit temp: 32  
Enter ending Fahrenheit temp: 40

| Temp F | Temp C |
|--------|--------|
| 32     | 0.000  |
| 33     | 0.556  |
| 34     | 1.111  |
| 35     | 1.667  |
| 36     | 2.222  |
| 37     | 2.778  |
| 38     | 3.333  |
| 39     | 3.889  |
| 40     | 4.444  |

CMPS 150 – Fall 2005 -- Programming Assignment #6

1. Exponentiation
2. Print and Sum Multiples
3. Fahrenheit / Celsius Table
4. Count Characters
5. Quit

Choice: 4

Enter the sentinel string value (NO spaces): quit

Enter a word (NO spaces): this

Enter a word (NO spaces): is

Enter a word (NO spaces): a

Enter a word (NO spaces): wonderful

Enter a word (NO spaces): assignment

Enter a word (NO spaces): quit

The sum of all characters entered: 26

1. Exponentiation
2. Print and Sum Multiples
3. Fahrenheit / Celsius Table
4. Count Characters
5. Quit

Choice: 7

Invalid Selection -- Please Try Again !

1. Exponentiation
2. Print and Sum Multiples
3. Fahrenheit / Celsius Table
4. Count Characters
5. Quit

Choice: 5

Thank you for using the program ... Goodbye !!

**4) Additional Requirements:**

- Use comments as appropriate. Refer to the “Programming Style Sheet” on the CMPS 150 web site.
- Your program must use good names for all variables and named constants. (Good names are names that are descriptive of the values stored or the function performed.)
- Adhere to style requirements. See “Programming Style Sheet” on the CMPS 150 web site.

**5) Name your source file 'pa6.cc' and store it in your class directory (cs150x).**

**6) Compile your program and test it (see Some Unix Help for quick assistance).**

To compile:

```
g++ -o pa6run pa6.cc
```

To run (execute):

```
pa6run
```

**7) After it is debugged and running correctly, submit pa6.cc (the source file only) electronically by 10:00 PM, Tuesday, October 25, 2005 to receive full credit.**

```
submit -d pa6.cc
```

The CLID for the TA of your section is the name to put in. This is one of the following:

| <u>Section</u> | <u>TA</u>   | <u>CLID of TA</u> |
|----------------|-------------|-------------------|
| Section 3..... | Gesan ..... | gxw2096           |
| Section 4..... | Anca.....   | axd9917           |
| Section 5..... | Mitun ..... | mxb2169           |
| Section 6..... | Jason ..... | jbm8240           |

You will be asked to enter the assignment name and the CLID of the person it is to go to. The assignment name is:

```
assn6
```

REMINDER: You can turn in programs up to 24 hours late for 75% credit, or up to 48 hours late for 50% credit.

**NOTE: Programs that do NOT compile will receive a grade of zero !!!**