```
Script started on 2023-10-04 12:24:17-05:00 [TERM="xterm" TTY="/dev/pts/2" COLUMNS=
mf98604@ares:~$ pwd
/home/students/mf98604
mf98604@ares:~$ cat adjdicestat.info

Name: Philip May'r
Class: CSC121-001

Activity: Adjusted Dice Roll Statistics Program
Level: 12

Description: Determines dice roll statistics.

mf98604@ares:~$ show-code adjdicestat.cpp


adjdicestat.cpp:
```

```cpp
#include "random.h"
#include <iostream>
#include <cctype>
#include <string>

using namespace std;

void get_dice_roll(short & dice,
                   short & sides,
                   short & adjustment,
                   char & d,
                   char & plus_minus)
{
    cout << "\n\t\tWelcome to the Dice Statistics Programs!!!\n";

    cout << "\nEnter your dice roll: ";

    while (cin.peek() != '\n' && isspace(cin.peek()))
    {
        cin.ignore();
    }

    if (!isalpha(cin.peek()))
    {
        cin >> dice;
    }
        else
    {
        dice = 1;
    }

    cin >> d >> sides;

    while (cin.peek() != '\n' && isspace(cin.peek()))
    {
        cin.ignore();
    }

    if (cin.peek() != '\n')
    {
        cin >> plus_minus >> adjustment;
    }

    if (plus_minus == '-')
    {
        adjustment = -adjustment;
    }

    cout << "\n\nThank you!!  Calculating...  ";
}

void calculate_results(short dice,
                       short sides,
                       short adjustment,
                       short & min_roll,
                       short & max_roll,
                       short & random_roll,
                       double & average_roll)
{
    min_roll = dice + adjustment;
    max_roll = static_cast<short>((dice * sides) + adjustment);
    average_roll = (min_roll + max_roll) / 2.0;
    random_roll = 0;

    for (int i = 0; i < dice; i++)
    {
        random_roll += get_random_num(1, sides);
    }
    random_roll += adjustment;

    cout << "Done.\n";
}

void print_results(short dice,
                   short sides,
                   short adjustment,
                   short min_roll,
                   short max_roll,
                   short random_roll,
                   double average_roll,
                   char plus_minus)
{
    string die_dice;

    die_dice = (dice > 1) ? "dice" : "die";

    if (adjustment < 0)
    {
        adjustment = -adjustment; // plus_minus takes cares of the sign
```

```
    }

    cout << "\nWhen rolling " << dice << " " << sides << "-sided "
         << die_dice << " adjusted by " << plus_minus << adjustment
         << ", your statistics will be:\n"
            "\n\tMinimum:  " << min_roll
         << "\n\tAverage:  " << average_roll
         << "\n\tMaximum:  " << max_roll
         << "\n\nA typical dice roll might result in " << random_roll
         << ".";

    cout << "\n\nThank you for using the DSP!!"
            "\n\nHave a wonderful day!\n\n";
}

int main()
{
    short dice,
          sides,
          adjustment{0},
          min_roll,
          max_roll,
          random_roll;

    double average_roll;

    char d;
    char plus_minus;

    srand(static_cast<unsigned>(time(nullptr)));

    get_dice_roll(dice, sides, adjustment, d, plus_minus);
    calculate_results(dice, sides, adjustment, min_roll, max_roll,
                      random_roll, average_roll);
    print_results(dice, sides, adjustment, min_roll, max_roll,
                  random_roll, average_roll, plus_minus);

    return 0;
}
```

```
mf98604@ares:~$ show-code random.cpp
```

random.cpp:

```
#include "random.h"
#include <cstdlib>

short get_random_num(short lower_bound, short upper_bound)
{
    return static_cast<short>(rand() %
                              (upper_bound - lower_bound + 1) +
                              lower_bound);
}
```

```
mf98604@ares:~$ CPP adjdicestat random
adjdicestat.cpp***
random.cpp...


mf98604@ares:~$ CPP adjdicestat random
show-code adjdicestat.cpp
cat adjdicestat.info
pwdshow-code adjdicestat
cat adjdicestat.info
pwd./adjdicestat.out

                Welcome to the Dice Statistics Programs!!!

Enter your dice roll: d6


Thank you!!  Calculating...  Done.

When rolling 1 6-sided die adjusted by 0, your statistics will be:

        Minimum:  1
        Average:  3.5
        Maximum:  6

A typical dice roll might result in 4.

Thank you for using the DSP!!

Have a wonderful day!

mf98604@ares:~$ ./adjdicestat.out

                Welcome to the Dice Statistics Programs!!!

Enter your dice roll: 2d6


Thank you!!  Calculating...  Done.

When rolling 2 6-sided dice adjusted by 0, your statistics will be:

        Minimum:  2
        Average:  7
        Maximum:  12

A typical dice roll might result in 9.

Thank you for using the DSP!!

Have a wonderful day!

mf98604@ares:~$ ./adjdicestat.out
```

```
                Welcome to the Dice Statistics Programs!!!

Enter your dice roll: 2d6+2


Thank you!!  Calculating...  Done.

When rolling 2 6-sided dice adjusted by +2, your statistics will be:

        Minimum:  4
        Average:  9
        Maximum:  14

A typical dice roll might result in 12.

Thank you for using the DSP!!

Have a wonderful day!

mf98604@ares:~$ ./adjdicestat.out

                Welcome to the Dice Statistics Programs!!!

Enter your dice roll: 2d6-2


Thank you!!  Calculating...  Done.

When rolling 2 6-sided dice adjusted by -2, your statistics will be:

        Minimum:  0
        Average:  5
        Maximum:  10

A typical dice roll might result in 1.

Thank you for using the DSP!!

Have a wonderful day!

mf98604@ares:~$ ./adjdicestat.out

                Welcome to the Dice Statistics Programs!!!

Enter your dice roll:    D   6


Thank you!!  Calculating...  Done.

When rolling 1 6-sided die adjusted by 0, your statistics will be:

        Minimum:  1
        Average:  3.5
        Maximum:  6

A typical dice roll might result in 2.

Thank you for using the DSP!!

Have a wonderful day!

mf98604@ares:~$ cat adjdicestat.tpq

Thought Provoking Questions - Lab 10 - Adjusted Dice Roll Statistics

1.)

cin.peek() can be used to decide whether an adjustment is included in
the input by looking for a newline character after the initial roll value.

2.)

The value of the plus or minus sign variable can simply be printed along with
the output. No decision-making is necessary.

3.)

The average formula is not affected by the adjustment factor because
the difference between the minimum and maximum values does not change.
With or without adjustment, the difference between
the minimum and maximum values remains the same.

4.)

A positive number added to a positive number remains positive.
A negative number added to a positive number is the same as
as subtracting that number less the negative sign from the positive number.
For that reason a plus sign can be used in both cases.

mf98604@ares:~$ exit
exit

Script done on 2023-10-04 12:25:44-05:00 [COMMAND_EXIT_CODE="0"]
```