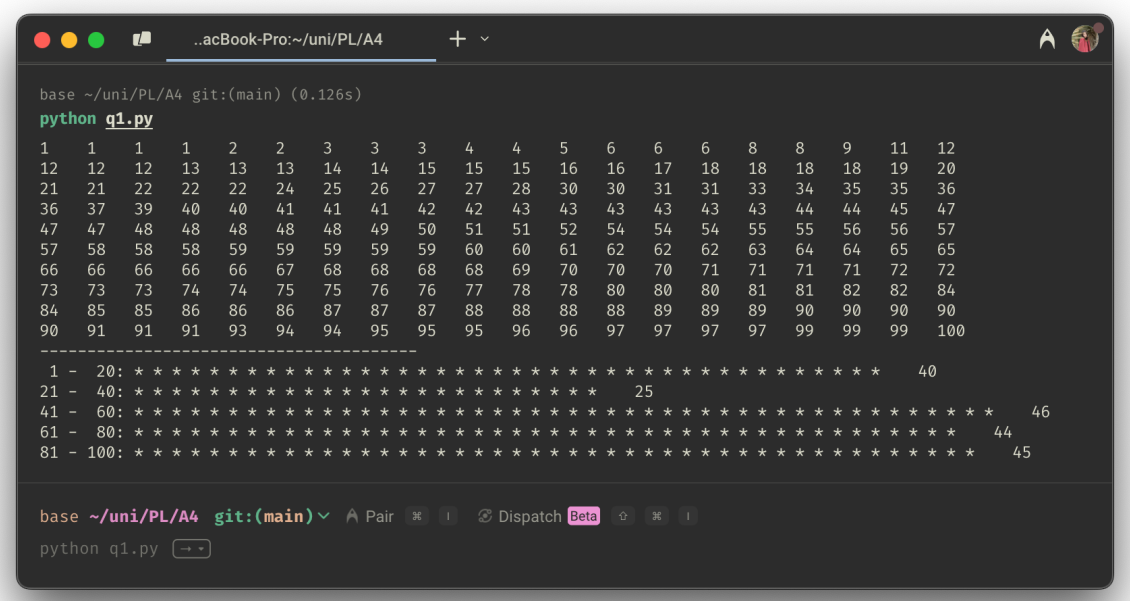
Programming Languages
Assignment 4

Spring 2025

1. Create a RandomStats.py application that generates 200 random numbers between 1 and 100, and then sorts them. The program should also make use of a separate array of 5 counters to count how many numbers fall into which category: 1 to 20, 21 to 40, 41 to 60, 61 to 80, and 81 to 100. A "star-graph" will print a number of stars to represent how many of each number were in any given category. Application output should look similar to:

```
   1   1   2   2   3   3   4   4   5   5   6   7   7   7   7   8   9   9   9   9
  10  11  11  13  13  13  13  14  14  14  14  15  15  16  16  16  17  18  19  19
  21  22  23  24  24  25  25  25  26  26  28  29  29  29  29  30  30  30  30  31
  31  32  32  32  32  32  34  34  35  37  37  37  38  38  39  40  41  41  41  41
  41  42  43  43  43  44  44  44  44  44  44  45  45  45  47  47  47  48  48  49
  49  49  51  51  51  51  51  52  52  53  53  54  54  54  55  55  56  56  56  57
  58  58  58  61  61  61  61  62  63  63  63  64  64  64  65  65  65  67  68  68
  69  71  71  71  71  72  72  73  73  73  74  74  75  75  76  77  77  78  79  80
  80  80  82  82  84  86  86  86  87  87  87  88  88  89  89  90  91  91  93  93
  93  93  93  94  94  95  95  95  96  96  97  98  98  98  98  98  98  99 100 100
--------------------------------------
 1 -  20: ****************************************   40
21 -  40: ************************************   36
41 -  60: ***********************************************   47
61 -  80: ***************************************   39
81 - 100: **************************************   38
```

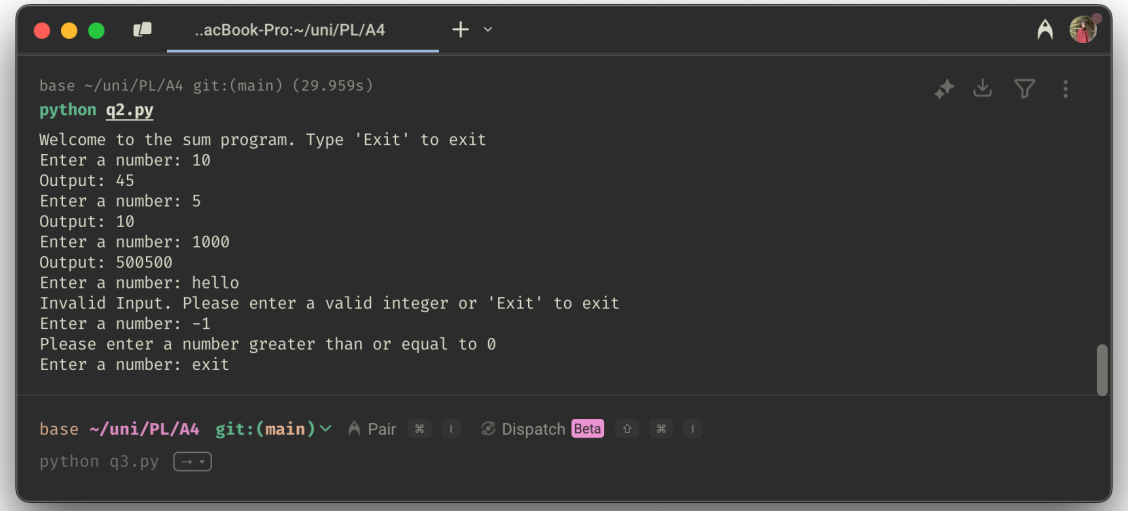
Output:



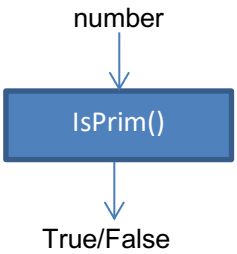
```
base ~/uni/PL/A4 git:(main) (0.126s)
python q1.py
1    1    1    1    2    2    3    3    3    4    4    5    6    6    6    8    8    9    11   12
12   12   12   13   13   13   14   14   15   15   15   16   16   17   18   18   18   18   19   20
21   21   22   22   22   24   25   26   27   27   28   30   30   31   31   33   34   35   35   36
36   37   39   40   40   41   41   41   42   42   43   43   43   43   43   43   44   44   45   47
47   47   48   48   48   48   48   49   50   51   51   52   54   54   54   55   55   56   56   57
57   58   58   58   59   59   59   59   59   60   60   61   62   62   62   63   64   64   65   65
66   66   66   66   66   67   68   68   68   68   69   70   70   70   71   71   71   71   72   72
73   73   73   74   74   75   75   76   76   77   78   78   80   80   80   81   81   82   82   84
84   85   85   86   86   86   87   87   87   88   88   88   88   89   89   89   90   90   90   90
90   91   91   91   93   94   94   95   95   95   96   96   97   97   97   97   99   99   99   100
----------------------------------------
 1 -  20: * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    40
21 -  40: * * * * * * * * * * * * * * * * * * * * * * * * *    25
41 -  60: * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    46
61 -  80: * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    44
81 - 100: * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    45

base ~/uni/PL/A4  git:(main)
python q1.py
```

Q2: Write a program that prompts the user to insert a number n or “Exit” and calculates the sum of all numbers before the inserted number. For example, if the input is 5, then the sum will be 1+2+3+4 =10
To do that, implement a recursive Python function that returns the sum of the first n integers.

Output:



Q3: As you know, the first six prime numbers are: 2, 3, 5, 7, 11, and 13, where the 6th prime is 13. Write a program that prompts the user of the prime rank.



For example: What is the prime number at rank: **6**

Then you program will return 13.

Hint: to find the prime number at rank n, you program needs to search all integer numbers from the first prime number until it finds the nth prime number.
Thus, you can implement a help function called isPrime() to test whether the input number is prime or not.

Output:
What is the prime number at rank: **6**
The prime number is 13

What is the prime number at rank: **10**
The prime number is 29

Output: