### **Program 1:**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Student grades  -1 | Processing items: totalScore, i  Algorithm:  1. User enters student grades one at a time  2. while (input != -1)  3. if input is not -1 then each student grade is added to totalScore, and i is incremented  (user will enter -1 to break the while loop)  4. endwhile  5. output average score (totalScore / i)  6. output number of scores (i) | The average student score and number of scores |

### **Program 2:**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Pledges or -1 | Processing items: none  Algorithm:  1. user enters pledges one at a time. User enters -1 to terminate data entry  2. while (input != -1)  3. if (input >= 100) pledgesOver100++  4. endwhile  5. output pledgesOver100 | The number of pledges greater than or equal to $100 |

### **Program 3:**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Votes for candidates (by first letter [a,A b,B] | Processing items: none  Algorithm:  1. User enters votes one at the time. Data entry is terminated with q  2. while (input != q)  3. input = tolower(input)  4. if input == a then increment votesForAndy  5. if input == b then increment votesForBeth  6. endwhile  7. output the number of votes andy got, and the number of votes beth got  8. if votesForAndy == votesForBeth, tell the user it was a tie  9. if votesForAndy > votesForBeth tell the user andy won  10. else tell the user that beth won | The number of votes for each candidate and who won |

### **Program 4:**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Combos ordered | Processing items: none  Algorithm:  1. User enters order by combo code (a,b,c). input is terminated with q (or Q)  2. while (input != q)  3. input = tolower(input)  4. switch (input)  5. case a: increment comboA  6. case b: increment comboB  7. case c: increment comboC  8. endswitch  9. endwhile  10. display the number of each combo ordered  11. display the total price ((comboA \* 6.00) + (comboB \* 6.25) + (comboC + 5.75)) | Number of each combo ordered and the total price due |