Chapter 3: Variables and Assignment

1) The temperature of a special thermometer is worked out using the following formula:

\[ T = \frac{V - 3.5}{8.2} \times 100 \]

where \( V \) is the voltage. Write a program that inputs the voltage \( V \) and works out the temperature \( T \).

2) Write a program to input three numbers and print their average.

Enter three numbers (separated by comma):  5,20,23
The average is:  16

3) The ASCII code for each lower case letter is 32 higher than the corresponding upper case letter. Write a program that accepts from the keyboard a character \( ch \) and outputs a character with the ASCII code 32 higher than \( ch \).

Enter a character:  K
New character:  k

4) An object with initial velocity \( u \) accelerates at a rate of \( a \) m/s\(^2\) reach a final velocity \( v \). The distance \( s \) the object travels can be represented by a formula:

\[ s = \frac{(v^2 - u^2)}{2a} \]

Write a program to find the distance travelled with \( v, u \) and \( a \) as the user inputs.

5) Conversion form Fahrenheit to Celsius is given by the following formula:

\[ C = \frac{5}{9}(F - 32) \]

Write a program to input a temperature and convert it into Fahrenheit or Celsius temperature according to the user’s choice.
Chapter 5: Conditional Statements

1) Write a program to compare two input integers and print out the answers.

Enter number 1: 27
Enter number 2: 70
Number 1 is smaller than number 2.

Enter number 1: 14
Enter number 2: 14
They are equal.

2) Write a program to determine the grade of a student according to the student’s score in DSE as following:

<table>
<thead>
<tr>
<th>Mark</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 ≤ x ≤ 100</td>
<td>5**</td>
</tr>
<tr>
<td>80 ≤ x &lt; 90</td>
<td>5*</td>
</tr>
<tr>
<td>70 ≤ x &lt; 80</td>
<td>5</td>
</tr>
<tr>
<td>55 ≤ x &lt; 70</td>
<td>4</td>
</tr>
<tr>
<td>40 ≤ x &lt; 55</td>
<td>3</td>
</tr>
<tr>
<td>30 ≤ x &lt; 40</td>
<td>2</td>
</tr>
<tr>
<td>20 ≤ x &lt; 30</td>
<td>1</td>
</tr>
<tr>
<td>0 ≤ x &lt; 20</td>
<td>UNCL</td>
</tr>
</tbody>
</table>

Enter test score: 79.9
Grade: 5

Enter test score: 80
Grade: 5*

Enter test score: 101
Error.
3) Write a program to solve the quadratic equation in the form \( ax^2 + bx + c = 0 \). The coefficients \( a, b \) and \( c \) are input from the keyboard. Your program should consider the following cases:

1. When both \( a \) and \( b \) are zero, output "It is not an equation."

2. When \( a = 0 \) and \( b \neq 0 \), \( x = -\frac{c}{b} \).

3. When \( a \neq 0 \) and \( b^2 - 4ac < 0 \), output "There is no solution."

4. When \( a \neq 0 \) and \( b^2 - 4ac = 0 \), \( x = -\frac{b}{2a} \).

5. When \( a \neq 0 \) and \( b^2 - 4ac > 0 \), \( x = -\frac{b\pm\sqrt{b^2-4ac}}{2a} \).

4) Set up a simple multiple choice quiz, then give out the answer when the users have finished the input after each question, for example:

Quiz

Question 1: How many stations are there on the MTR West Rail Line?
(A) 9  (B) 10  (C) 11  (D) 12
Your answer: D
Correct!

Question 2: What is the next station of Shatin (on East Rail Line)?
(A) Kowloon Tong  (B) Kowloon  (C) Kowloon Bay  
(D) Not a good question, you don’t state which way you are going.
Your answer: D
Correct! There are 3 "next" stations: Fo Tan, Tai Wai and Racecourse.

Question 3: There are 10 sheeps getting on the MTR train in Lo Wu, and then 2 wolves also enter the train in Sheung Shui. Assume that the wolves are so hungry that each of them can eat 1 sheep before the train stops at the next MTR station. How many animals are left when the train arrives at Tai Po Market Station?
(A) 2  (B) 12  (C) 6  (D) 0
Your answer: C
Incorrect! The answer is D. No animals are allowed to enter MTR stations.

..................
5) Consider there are 4 restaurants in Wo Che: Itamomo, Maxim’s, Mikichi Ramen (味吉拉麵館), and JUSCO. The minimum cost needed to have a lunch in these restaurants are shown as following:

<table>
<thead>
<tr>
<th>Restaurant</th>
<th>Min. cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itamomo</td>
<td>$30</td>
</tr>
<tr>
<td>Maxim’s</td>
<td>$25</td>
</tr>
<tr>
<td>Mikichi</td>
<td>$55</td>
</tr>
<tr>
<td>JUSCO</td>
<td>$15</td>
</tr>
</tbody>
</table>

Write a program to give suggestions of which restaurant(s) can be chosen for lunch according to the amount of money in the wallet.

How much money is in your wallet now?  $30
You can have lunch at Itamomo, Maxim’s and JUSCO.

How much money is in your wallet now?  $10
You do not have enough money to have lunch at any of 4 restaurants.
Chapter 6: Repetition in Programs

1) Write a program to accept two numbers from 1 to 100 (you can assume that only numbers are entered) and find the quotient and remainder when the larger number is divided by the smaller number.

Enter number 1: 101
Error. Please re-enter.
Enter number 1: 8
Enter number 2: -1
Error. Please re-enter.
Enter number 2: 2934872309482
Error. Please re-enter.
Enter number 2: 70
The quotient of 70 divided by 8 is 8 and the remainder is 6.
Enter number 1: 14
Enter number 2: 14
They are equal.

2) The LCM (Lowest Common Multiple) of two numbers x, y is the smallest whole number which is divisible by x and y. The following algorithm is one of the usable ways to find LCM, but not the most efficient one.

<table>
<thead>
<tr>
<th>Step 1: Store x*y to m.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2: If m is divisible by both x and y, store m to LCM.</td>
</tr>
<tr>
<td>Step 3: Decrease m by 1.</td>
</tr>
<tr>
<td>Step 4: Repeat steps 2 and 3 as long as m is greater than or equal to both x and y.</td>
</tr>
<tr>
<td>Step 5: Output LCM.</td>
</tr>
</tbody>
</table>

Write a program to input two positive numbers and find their LCM. Your program should give the following sample output:

Enter two numbers: 3 4
The LCM of 3 and 4 is 12
3) The HCF (Highest Common Factor) of two numbers \(x, y\) is the largest whole number which can divide \(x\) and \(y\). The following algorithm is one of the usable ways to find HCF, but not the most efficient one.

<table>
<thead>
<tr>
<th>Step 1: Store 1 to (f).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2: If both (x) and (y) are divisible by (f), store (f) to HCF.</td>
</tr>
<tr>
<td>Step 3: Increase (f) by 1.</td>
</tr>
<tr>
<td>Step 4: Repeat steps 2 and 3 as long as (f) is smaller than or equal to the smaller value among (x) and (y).</td>
</tr>
<tr>
<td>Step 5: Output HCF.</td>
</tr>
</tbody>
</table>

a) Write a program to input two positive numbers and find their HCF. Your program should give the following sample output:

Enter two numbers: 162 729
The HCF of 162 and 729 is 81

b) Print a table of prime numbers from 1 to 200 (prime numbers: a number (larger than 1) that is not divisible by any number except 1).

<table>
<thead>
<tr>
<th>2</th>
<th>3</th>
<th>5</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>13</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>23</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>43</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>73</td>
<td>79</td>
<td></td>
</tr>
</tbody>
</table>

4) Write a program to allow players to guess a number between 1 to 100 (not including 1 and 100). The range narrows when players make a wrong guess.

1 to 100
Make a guess: 50

50 to 100
Make a guess: 25
Make a guess: 75

50 to 75
Make a guess: 63
Bingo!
5) Write programs to finish following tasks:
   a) Write a program to simulate the game of tossing a coin. Ask the user to guess head or tail by pressing the key ‘H’ or ‘T’ respectively. Show the result of the game with an appropriate message. Repeat the game until the user presses a key other than ‘H’ and ‘T’.

   Enter your guess: H
   You win!
   The outcome is H

   Enter your guess: T
   You lose!
   The outcome is H

   Enter your guess: E
   Thank you for trying!

   b) Write a program to simulate the game of comparing poker cards. First draw one random poker card. Then ask the user to guess if next the poker card is bigger, smaller than or equal to the previous card. The ASCII codes of diamond, club, heart and spade are as follow:

   Consider Diamond A is the smallest and the Spade K is the biggest.
Chapter 7: Arrays

1) Write a program to generate a randomized deck of playing cards. Use the following algorithm:
   1. Declare an array with 52 elements of integer, representing 52 cards in the deck.
   2. Initialise the array. Each element contains a value equivalent to the array index.
   3. Shuffle the array by swapping any two randomly picked numbers for 200 times.
   4. Print the first 5 elements.
   (You may further use 2D array to make a deck of cards in four suits, each with 13 cards.)

2) A palindrome is a word, phrase or number which reads the same, either backward or forward. For example, 25352 is a palindrome but 542 is not.
   Write a program to input a number and test whether it is palindrome. You should store the digits in an array. Then, compare the digits from both ends of the array.

3) Two dice are rolled and the possible results are:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

Write a C program to simulate the rolling of two dice for 100 times. Accumulate the frequency of each outcome and display the results using a histogram of “*” as follows:

```
1 2 3 4 5 6
1 2 3 4 5 6
2 3 4 5 6 7
3 4 5 6 7 8
4 5 6 7 8 9
5 6 7 8 9 10
6 7 8 9 10 11
```

Please press any key to continue …
Chapter 8: Strings

1) Write a program to construct a random sentence. A sentence consists of the following in order:

   Article + noun + verb + preposition + article + noun +.

Your program should randomly pick appropriate words and combine them to form a sentence. Declare four 2D arrays of characters to store the words shown in the following table first:

<table>
<thead>
<tr>
<th>Article</th>
<th>a</th>
<th>no</th>
<th>the</th>
<th>one</th>
<th>some</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>boy</td>
<td>ball</td>
<td>cat</td>
<td>dog</td>
<td>car</td>
</tr>
<tr>
<td>Verb</td>
<td>played</td>
<td>jumped</td>
<td>ran</td>
<td>climbed</td>
<td>shouted</td>
</tr>
<tr>
<td>Preposition</td>
<td>to</td>
<td>on</td>
<td>in</td>
<td>with</td>
<td>over</td>
</tr>
</tbody>
</table>

(Extra: Add an ‘s’ after the noun if the article is “some” to make it plural.)

2) Write a program to do a “rearranging sentence” (重組句子) exercise. You may use the sentences from the following table:

| 1. My favourite colour is orange. |
| 2. When can the wood laugh? |
| 3. I looked for Mary at the bus station. |
| 4. Harry Potter was a highly unusual boy in many ways. |
| 5. Her eyes were red from crying. |
| 6. The leaves will shake loose from the trees and fall. |
| 7. Sam is an architect. |

a) The program should be able to:
   1. Break a sentence into words and store them into a 2D array in lowercase letters. You may keep or abandon the punctuation.
   2. Randomly swap the words for 100 times.
   3. Print out the words and accept a string input.
   4. Compare the input string with the answer.
   5. Loop steps 3 and 4 until the user gets the answer correctly.

   Q: favourite / orange / my / . / is / colour
   A: My favourite orange is colour.
   A: my favourite colour is orange
   A: My favourite colour is orange.
   Correct!

b) Improve the program so that it will pick a sentence randomly to start.
3) Write a program to perform a data type check in order to ensure the input is a numeric string. The function atoi() is needed. If the conversion is successful, the function will return the number in an integer format. If the conversion fails, it will return a zero value. Loop until the input is a valid numeric string.

4) Write a hangman game which is a word-guessing game.
   1. Declare a 2D array with 10 words (such as “beautiful”, “happy”)
   2. Randomly pick a word from the 2D array as the answer key.
   3. Create a clue with all “*”, where the length is equal to that of the answer key.
   4. Show the clue and ask the user to input a letter for the guess.
   5. Check whether the letter exists in the answer key.
   6. Replace the “*” with the letter if the letter exists.
   7. Loop steps 4 to 6 until the answer key is guessed.
   (Extra: Count the lives and the game is over after several unsuccessful trials.)
Chapter 9: User Defined Functions

1) Write a program to play a game as follow:

   Decide the control buttons (up, down, left, right) and the program should reject error inputs. Define at least one new function.

2) Write a game program to accept 4 players to play a game as follow:

   1. Initially every player has 500 points.
   2. A player rolls 2 dices.
   3. If the sum of dices equals to—
      - 7: +100 points
      - 8: +50 points
      - 9: -100 points
      - 10: Game over
   4. Next player will roll the dices again and repeat step 3.
   5. There are 2 conditions those will cause a player’s game over:
      1. Get a ‘10’ when rolling dices
      2. The points drop to 0
   6. When 3 of 4 players’ games are over, the remaining player wins the game.

Define at least one new function. No global variables are allowed.
3) Write a program to allow user to do the following tasks:
   a) Change a word: delete the word in the sentence and paste the new word.
   b) Delete a word from the array.
   c) Add a word after a selected word.

In this program, you are not allowed to break the 1-D array sentence into a 2-D array of words (like 8.2).

*No global variables are allowed.*
1) This is a text file which contains the result of Boys Grade B 100m final:

<table>
<thead>
<tr>
<th>Array</th>
<th>cls</th>
<th>name</th>
<th>Time</th>
</tr>
</thead>
</table>
| 0     | 6A   | Tom    | 10'54"
| 1     | 6B   | Jerry  | 16'39"
| 2     | 6C   | Peter  | 08'62"
| 3     | 6D   | Leon   | 30'38"
| 4     | 6A   | Richard| 14'44"
| 5     | 6D   | Wilson | 08'70"
| 6     | 6E   | Galileo| 29'00"
| 7     | 6A   | Bryan  | 15'03"

a) Write a program to read the text file and separate the content into 3 arrays: “cls”, “name” and “time” as follow:

b) Construct a new array called “rank” to save the rank of each competitor, and print out the result in the order of rank. You may use min[] and sec[] to separate the time in the array “time”.

c) Rearrange the order of all records to print out the result in the order of rank by bubble sort (i.e. You are not allowed to use array “rank”, but can use min[] and sec[]).

Please assume that: (1) all the participants finish the race before 99'99”

(2) all the participants finish at different times

but do not assume the whole file has 8 records only.
2) Consider the following text file (left).

a) Write a program to rewrite the file to the one on the right:

![Original text](image1)

b) Write a program to rewrite the file to the one on the right:

![Original text](image2)