

The game of Black Jack

Introduction

In the game of Black Jack, ordinary playing cards are used. Each player is dealt two cards. He/she then asks for another card, and another, and so on, until he/she decides to stop. His/her objective is to obtain as high a total as possible, without exceeding 21. Face cards (i.e. Jack, Queen and King) are counted as 10, and all other cards counted at their face values. An Ace is counted as 11 if the total score obtained before is less than 11, otherwise it is counted as 1. When he/she decides to stop, he/she stands. If he/she goes over 21, he/she is bust.

The aim of this project is to simulate a game between the computer (which is also the dealer) and one player. Firstly, two cards and one card are dealt to the player and the computer respectively. Then the player is asked to have more cards until he/she is bust or stands. If the player is bust, the computer wins. If the player stands, cards are dealt to the computer. The strategy of the computer is to stand on any score of 17 or more. If the computer goes bust, the player wins. Otherwise, the one with a higher score wins. However, if the player and the computer have the same score, it will then come to a draw.

Program specifications

The following constant declaration may be used:

```
const Number = 'A23456789TJQK'; Suit = 'SHCD';
```

In the declaration, `Number[10] = 'T'` is used to represent the face value of 10. `Suit[1] = 'S'` is used to represent 'Spade'. `Suit[2] = 'H'` is used to represent 'Heart'. `Suit[3] = 'C'` is used to represent 'Club' and `Suit[4] = 'D'` is used to represent 'Diamond'.

You should write a program which does all of the following:

- (1) Create the 52 cards, AS, 2S, 3S, ... , QD, KD, by using arrays `Number[]` and `Suit[]`. Store the result into an 52-element string array `Card[]`.
- (2) Use a procedure to generate a random integer ranging from 1 to 52 inclusively. Use this random integer to select an element from the array `Card[]`. Since no repetition is allowed, use the Boolean array `Use[K]`, where $K = 1, 2, 3, \dots, 52$, to indicate that K is the chosen element if `Use[K]` is true or K is not the chosen element if `Use[K]` is false.
- (2) Randomly select two elements and one element from the array `Card[]` for the player and the computer respectively. Display them on the VDU.
- (3) Use a procedure to calculate the score value of each card. When an Ace is selected, count it as 11 if the total score obtained before is less than 11, otherwise count it as 1.
- (4) Unless the total score exceeds 21, deal more cards to the player until he/she decides to stop. Each time, remember to update his/her total score.
- (5) If the player goes bust, announce that the computer is the winner.
- (6) If the player stands, deal cards to the computer until the score reaches 17 or more. If the computer goes bust, announce that the player is the winner.
- (7) Display all cards dealt on the screen and the total scores obtained. Determine who is the winner and display the result on the VDU.
- (8) Repeat the game until the player wants to stop.

A sample output of this program runs as follows:

Player's cards: 3S AH
Player's total score = 14
Computer's card: TH
Computer's score: 10
Do you want another card (y/n)? y

Additional card for player: AC
Player's total score = 15
Do you want another card (y/n)? y

Additional card for player: 3C
Player's total score = 18
Do you want another card (y/n)? n

Additional card(s) for computer: 4C KS
Computer's total score = 23

Computer goes bust. Player wins!

Do you want another game (y/n)? y

Player's cards: TD 3H
Total score = 13
Computer's card: AS
Computer's score: 11
Do you want another card (y/n)? y

Additional card for player: KS
Total score = 23

Player goes bust. Computer wins!

Do you want another game (y/n)? n

Player's cards: AC JD
Total score = 21
Computer's card: 5S
Computer's score: 5
Do you want another card (y/n)? n

Additional card(s) for computer: 8S AD 5D
Computer's total score = 19

Player wins!

Do you want another game (y/n)? y

Player's cards: 7H 9D
Total score = 16
Computer's card: 2H
Computer's score: 2
Do you want another card (y/n)? n

Additional card(s) for computer: TS 7C
Computer's total score = 19

Computer wins!

Do you want another game (y/n)? n