1.Bonus scripting

Write a Linux Bash shell script to compute the bonus for salespersons who are working at Mercedes Benz dealership who sell the following models:

A class - £24,095 - £38,095 (average £31,095)

B Class - £28,045 - £38,280 (average £33,162)

C Class - £34,670 - £50,405 (average £42,537)

E Class - £39,680 - £69,015 (average £54,437)

AMG C65 - £78,103 - £81,217 (average £79,660)

Each salesperson earns a basic salary of £2,000 per month. However:

• If he or she achieves £200,000 worth of sales they will earn a bonus of £10,000 per month.

• If he or she sells more than £300,000 worth of sales they will earn a bonus of £15,000 per month.

• If he or she sells more than £400,000 worth of sales they will earn a bonus of £20,000 per month.

• If he or she sells more than £500,000 worth of sales they will earn a bonus of £25,000 per month.

• If he or she sells more than £650,000 worth of sales they will earn a bonus of £30,000 per month.

Design a program which calculates, each salespersons salary per month.

Calculate the tax each salesperson pays and hence calculate their total salary after tax.

Inputs

• Enter Month

• Enter Salespersons name

• Models he/she sold

• Save the data

o File name

o File path

Outputs

• How to display results

o Name order

o Salary

o Name with associated net salary

The program should work for between 3 and 20 salespersons.

Sorting

You should employ a bubblesort (do not use the bash sort command) to display a list of the SalesPersons. All inputs should be checked for valid data.

Tax Rates

Personal allowance- £12,500 – this is the amount you can earn before you start paying tax.

Basic Rate - £12,501 to £50,00 is taxed at 20%

Higher Rate- £50,001 to £150,000 is taxed at 40%

Remember to use regular expressions for error checking. i.e when you prompt the user to enter a value you should test to see if it a number/s or character/s are in the correct format.

When this is submitted you will need to include the following:

• Program code with comments

• Flowcharts

• User guide

• Description of how the code works – this is the implementation section (it is not a description of program when it is running).

• Evidence of testing

• Screenshots of outputs – this should show your personalised prompt

2. Blackjack Program

Write a shell script to play the card game Blackjack. There must be one dealer and at least one player. Name the program blackjack.sh.

This is the final program which will be included in the portfolio.

To get the highest mark the game should be as close as possible to the game described below. You will need to:

• use arrays and functions.

• determine how to ensure that a card which is dealt to one user is removed from the pack so that it is not available to other users.

• make the program user friendly.

Rules of the game

At the start of a blackjack game, the players and the dealer will receive two cards each. The players' cards are normally dealt face up, while the dealer has one face down (called the hole card) and one face up.

The aim of the game is to accumulate a higher point total than the dealer, but without going over 21. You compute your score by adding the values of your individual cards.

The cards 2 through 10 have their face value, J, Q, and K are worth 10 points each, and the Ace is worth either 1 or 11 points (player's choice).

The best possible blackjack hand is an opening deal of an ace with any ten-point card. This is called a "blackjack", or a natural 21, and the player holding this automatically wins unless the dealer also has a blackjack. If a player and the dealer each have a blackjack, the result is a push for that player. If the dealer has a blackjack, all players not holding a blackjack lose.

After the cards have been dealt, the game goes on with each player taking action - in clockwise order starting to dealer's left.

Each player can keep his hand as it is (stand) or take more cards from the deck (hit), one at a time, until either the player judges that the hand is strong enough to go up against the dealer's hand and stands, or until it goes over 21, in which case the player immediately loses (busts).

In most places, players can take as many cards as they like, as long as they don't bust (exceed 21).

When all players have finished their actions, either decided to stand or busted, the dealer turns over his hidden hole card.

If the dealer has a natural 21 (blackjack) with his two cards, he won't take any more cards. All players lose, except players who also have a blackjack, in which case it is a push - the bet is returned to the player.

If the dealer doesn't have a natural, he hits (takes more cards) or stands depending on the value of the hand. Contrary to the player, though, the dealer's action is completely dictated by the rules. The dealer must hit if the value of the hand is lower than 17, otherwise the dealer will stand.

If the dealer goes bust, all players who are left in the game win. Otherwise players with higher point totals than the dealer win, while players with lower totals or equal to that of the dealer lose.

Players with a blackjack win a bet plus a bonus amount, which is normally equal to half their original wager. A blackjack hand beats any other hand, also those with a total value of 21 but with more cards.

As described above, if the dealer has a blackjack, players with blackjack make a push, while all other players lose.

A points system could be used to determine an overall winner of the game.

Definitions

Push is when you and the dealer have the same hand value. When this happens your bet will be returned. You don't win anything and you don't lose anything.

Bust - having a total over 21, resulting in an automatic loss

Hit - To ask for another card. If that extra card makes the total over 21, then the player busts

Stick or stand - To stop asking for more cards

Five card trick/Charlie - Reaching five cards in one hand without busting; although increasingly uncommon, casinos may offer a 2:1 bonus for achieving it.

When this is submitted you will need to include the following:

• Program code with comments

• Flowcharts

• User guide

• Description of how the code works

• Evidence of testing

• Screenshots of outputs – this should show your personalized prompt