Group: 04

Name: Andalib Iftakher

ID: 22101348

Contribution Point: Genre Search, Binary Search Algorithm to find books in the database. Print an array of books.

Explanation:

1. First, I used to condition and prompts to make the user select a genre. I kept four genres and each genre is an array of books. Also to print the genres I ensured that each book will have a length of 15 characters (including price and name) this will enable me to use simple indexing to calculated the SI value for the first letter of the book or the last letter. For example, if the user wants to access the book at index 3, then the starting address will be 1E (each book is 0 to E characters long) so to find 1E we first decrease the value of the index by 1, we get 2, multiply with 10h, we get 20h, subtract the decreased index from 20h, we get 1Eh. Load it onto the SI register. And print using LEA instructions, while incrementing SI by 000Fh for every loop.
2. Then I used a binary search algorithm to find the books in the database. Since I needed to compare letters. During binary search I kept GR, GL and GM variables to keep track of my right, left and middle index. Then I utilized ordinary binary search algorithm just like in high level language.
3. The algorithm works this way: first it will ask the user to insert a character that they want to search, then it will be using binary search to locate any book starting with that character, if it finds any then it will ask the user to provide the other characters and as long as it matches it will not shift its GM value if it doesn’t, then I will keep the current SI - starting SI, value of the current book in a var and use binary search again. That value will enable us to compare the correct characters. For example, I have a book named JHONNY ENGLISH, and a book named JHONAH, say the input of the user was JHON and after that if they insert A then we’ll use binary search algorithm with SI (just first letter) + the value we calculated before.
4. I have asked the user to use block letters only and when the name of the book has been written then it will have to be terminated by an underscore so that the program knows that no more additional letters will be provided.

Name: Md. Nasimur Rahman

Id: 21201424

Contribution points:Creating cart and payment slip, a discount feature on number of items.

1. Explanation: Here a user can add one or multiple books to a cart so that he/she can check what books he/she added to purchase.
2. He/she could also add multiple books from multiple genres.
3. In the cart he/she is also able to see prices of every book he/she added to the cart.
4. Also there is a discount feature depending on how many books have been added such as if the user selects less than 2 books then there is no discount. If users add 3 or 4 books, users get a 10% discount on total price and on 5 or more

books users get a 20% discount on total price.

Name: Avinabo Tarafder

Id: 22101004

Contribution points: implement how print the cart(descending order) using sorting algorithm and calculated the total value from the strings of those array(extracting those num values)

Explanation:

I used a selection sort algorithm to reorder the array in a way that the prices of those num values are in descending order in the array. For example :

Before sort

CART DB "ALLEGIANT\_\_\_50$", "DIVERGENT\_\_\_60$", "INSURGENT\_\_600$"

After sort:

CART DB "INSURGENT\_\_600$", "DIVERGENT\_\_\_60$","ALLEGIANT\_\_\_50$"

Also in the meantime i extracted those num values like 50 , 60 ,600 and converted each of them to hexadecimal to add them   
  
And after printing the cart array the total which is hexadecimal is converted to decimal and then it is printed showing the whole cart

The CALL instruction is used whenever we need to make a call to some procedure or a subprogram. Whenever a CALL is made, the following process takes place inside the microprocessor: The address of the next instruction that exists in the caller program (after the program CALL instruction) is stored in the stack. The instruction queue is emptied for accommodating the instructions of the procedure. Then, the contents of the instruction pointer (IP) is changed with the address of the first instruction of the procedure. The subsequent instructions of the procedure are stored in the instruction queue for execution. The Syntax for the CALL instruction is as follows: The RET instruction stands for return. This instruction is used at the end of the procedures or the subprograms. This instruction transfers the execution to the caller program. Whenever the RET instruction is called, the following process takes place inside the microprocessor: The address of the next instruction in the mainline program which was previously stored inside the stack is now again fetched and is placed inside the instruction pointer (IP). The instruction queue will now again be filled with the subsequent instructions of the mainline program. When we are executing this line of code CALL HEX\_2\_DEC It takes goes to HEX\_2\_DEC After we are done with HEX\_2\_DEC it returns to where it was called from