Hashing Assignment

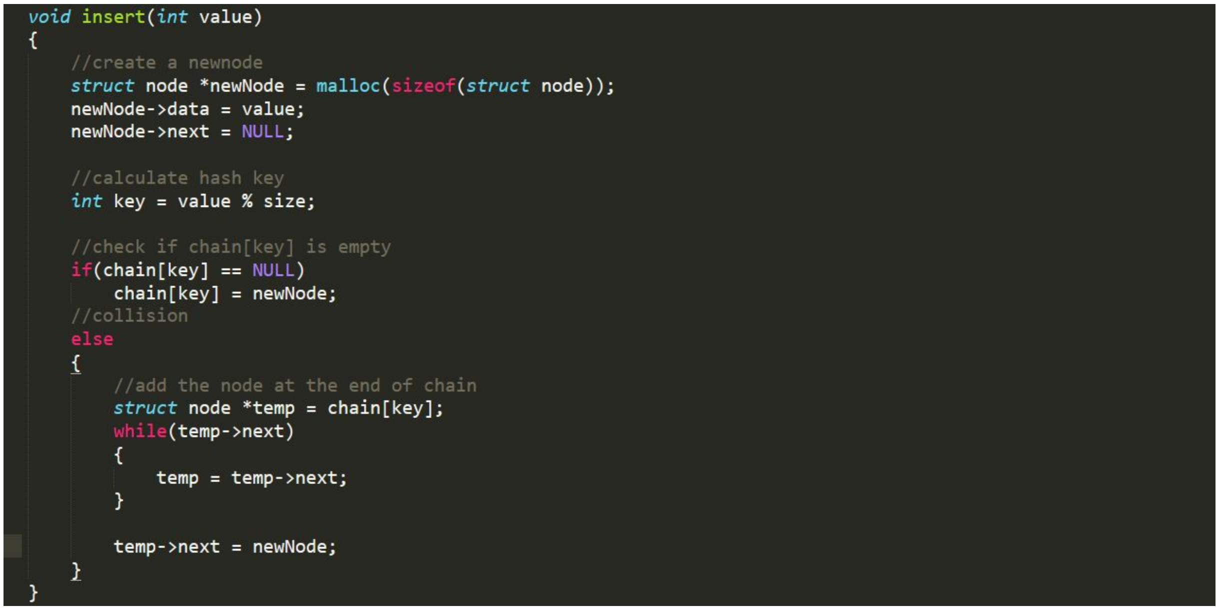
Data Structures and Algorithms II
IS 2110
Second Year - Group Assignment 01
Group 29

Name	**Index Number**	**Email Address**	**Contribution**
MR. HATHURUSINGHE H.B.P.	17020311	banuka96@gmail.com	Question 02
MS. PATHIRANA S.P.S.P.	17020654	shelomipriskila1995@gmail.com	Question 01
MS. WIJESINGHE G.L.P.	17020982	lahiruka.wijesinghe@gmail.com	Question 03

Question 01

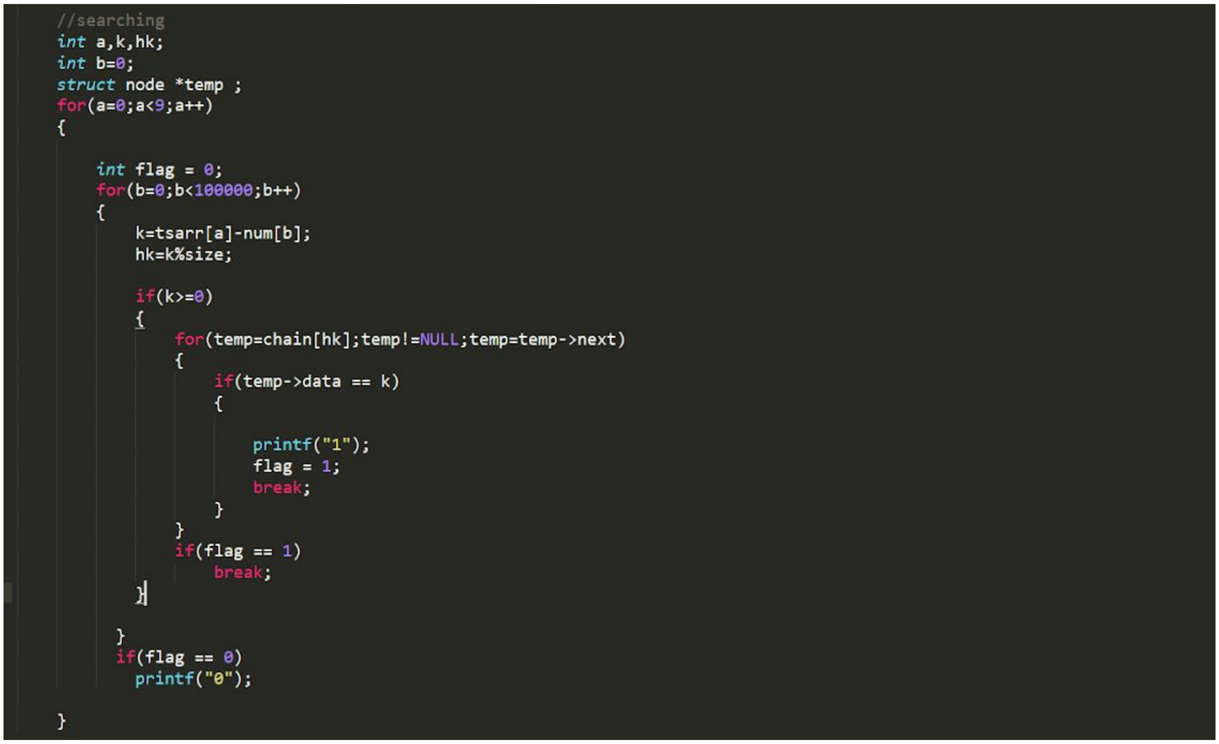
The program is built using the C language with chaining knowledge. First node is created with the pointer and then given size. Then there are separate functions for inserting value to hash table and printing hash table(if necessary). In the main function there is array called num with a size of 100000. Numbers from integers.txt file have read and taken to num array. Then insert function has called inside main function and inserted numbers to the hash table. Targeted sums are in the array called tsarr. Then there is searched function for getting nine bits. Inside the search function, num array subtracted from tsarr array using a for loop and named as k. Then k is hashed and named the result as hk. Then hk is searched in hash table. After finding hk, k is again searched through the linked list in the related hk. If its found 1 is printed for the related target sum, else 0.

Insert function



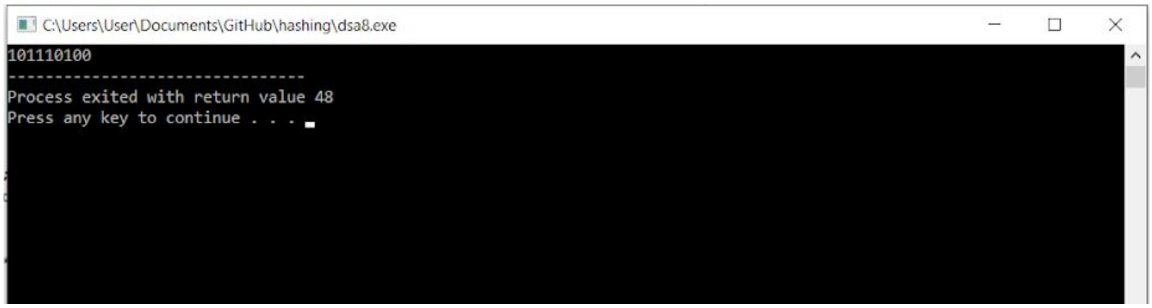
```
void insert(int value)
{
    //create a newnode
    struct node *newNode = malloc(sizeof(struct node));
    newNode->data = value;
    newNode->next = NULL;

    //calculate hash key
    int key = value % size;

    //check if chain[key] is empty
    if(chain[key] == NULL)
        chain[key] = newNode;
    //collision
    else
    {
        //add the node at the end of chain
        struct node *temp = chain[key];
        while(temp->next)
        {
            temp = temp->next;
        }

        temp->next = newNode;
    }
}
```

Searching function



```
//searching
int a,k,hk;
int b=0;
struct node *temp ;
for(a=0;a<9;a++)
{

    int flag = 0;
    for(b=0;b<100000;b++)
    {
        k=tsarr[a]-num[b];
        hk=k%size;

        if(k>=0)
        {
            for(temp=chain[hk];temp!=NULL;temp=temp->next)
            {
                if(temp->data == k)
                {

                    printf("1");
                    flag = 1;
                    break;
                }
            }
            if(flag == 1)
                break;
        }

      }
      if(flag == 0)
        printf("0");

}
```

Output



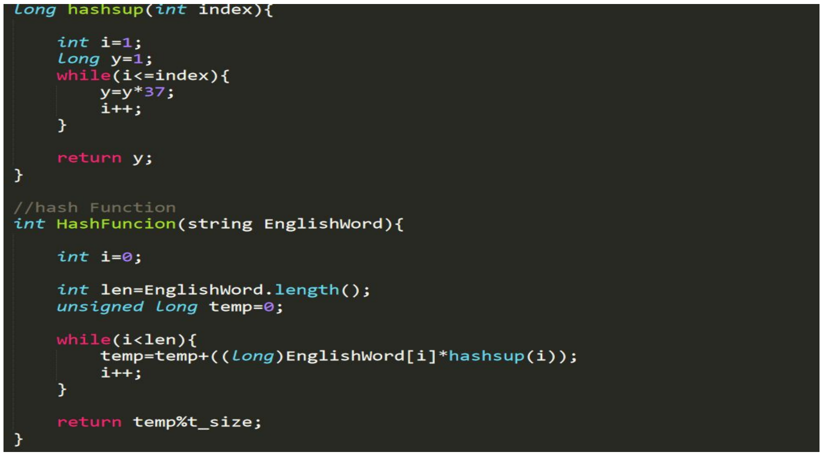
```
101110100
--------------------------------
Process exited with return value 48
Press any key to continue . . .
```

Sample Test Cases

No.	Test Case	Expected Output	Pass/Fail
1	Input integers.txt file	101110100 All have target sums except 2nd 6th 8th 9th.	Pass

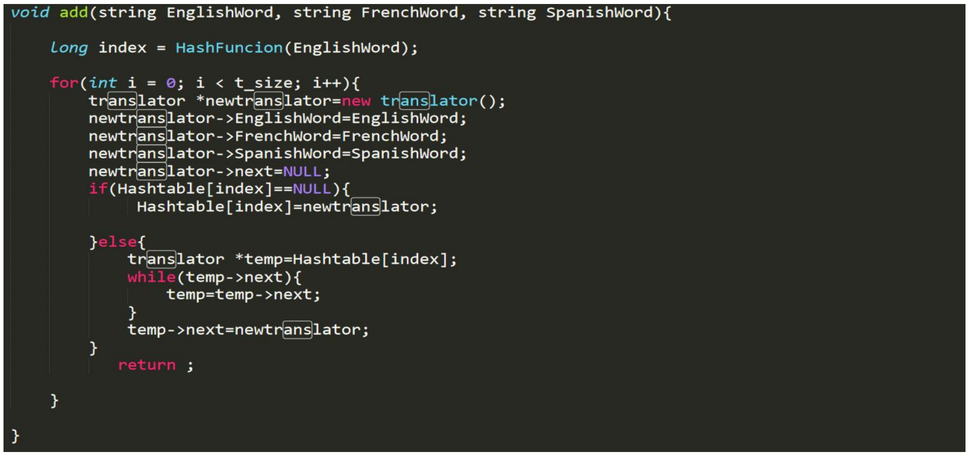
Question 02

In here there are key two documents. One is to read the words of each and every language(A) and the other document is the customer phrases included text file(B). Following Hash Function will store (A) words into a hash table.



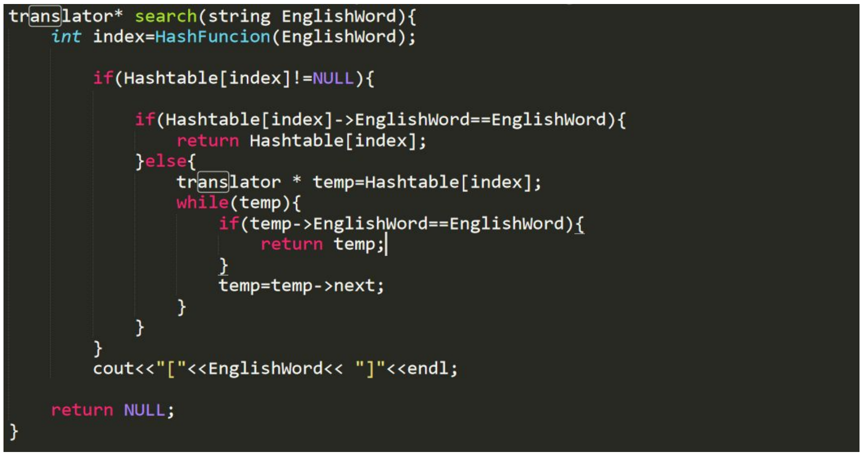
```
long hashsup(int index){

    int i=1;
    long y=1;
    while(i<=index){
        y=y*37;
        i++;
    }

    return y;
}

//hash Function
int HashFuncion(string EnglishWord){

    int i=0;

    int len=EnglishWord.length();
    unsigned long temp=0;

    while(i<len){
        temp=temp+((long)EnglishWord[i]*hashsup(i));
        i++;
    }

    return temp%t_size;
}
```

Storing these words into a hashtable node and a word from each three languages will placed under a hash key and those words will store like a linked list.



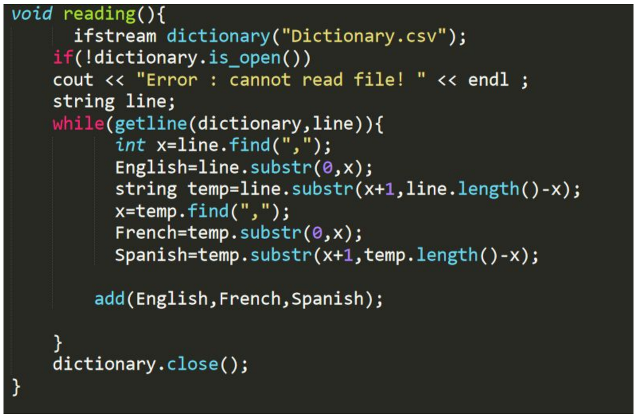
```
void add(string EnglishWord, string FrenchWord, string SpanishWord){

    long index = HashFuncion(EnglishWord);

    for(int i = 0; i < t_size; i++){
        translator *newtranslator=new translator();
        newtranslator->EnglishWord=EnglishWord;
        newtranslator->FrenchWord=FrenchWord;
        newtranslator->SpanishWord=SpanishWord;
        newtranslator->next=NULL;
        if(Hashtable[index]==NULL){
             Hashtable[index]=newtranslator;

        }else{
            translator *temp=Hashtable[index];
            while(temp->next){
                temp=temp->next;
            }
            temp->next=newtranslator;
        }
           return ;

    }

}
```

When we are inserting a particular word to search, it will automatically create a hash-key and go to the hash table and check whether that hash key is matching with the stored value under that hash key.

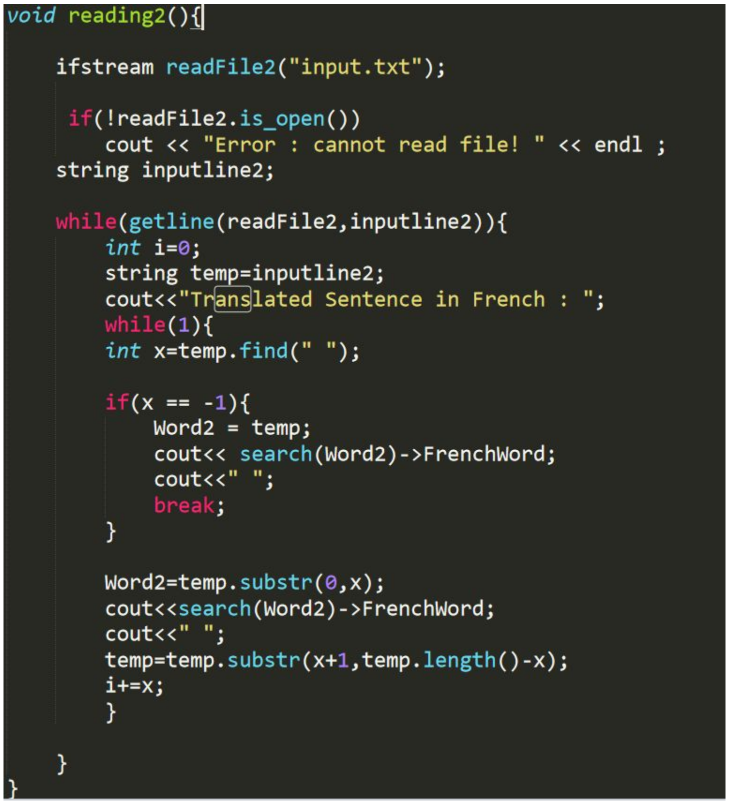


```
translator* search(string EnglishWord){
    int index=HashFuncion(EnglishWord);

        if(Hashtable[index]!=NULL){

            if(Hashtable[index]->EnglishWord==EnglishWord){
                return Hashtable[index];
            }else{
                translator * temp=Hashtable[index];
                while(temp){
                    if(temp->EnglishWord==EnglishWord){
                        return temp;
                    }
                    temp=temp->next;
                }
            }
        }
        cout<<"["<<EnglishWord<< "]"<<endl;

    return NULL;
}
```

Following functions are reading A and B Documents.

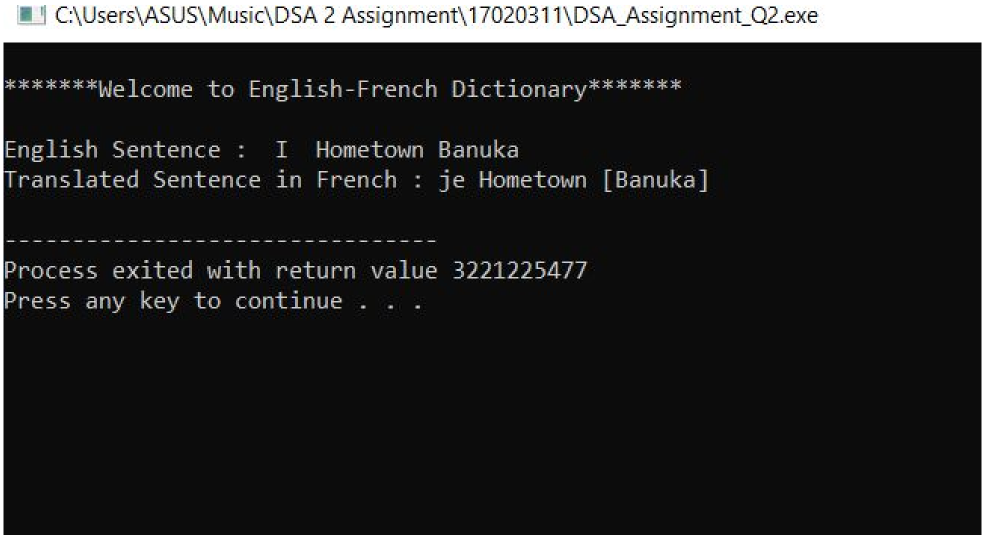


```
void reading(){
      ifstream dictionary("Dictionary.csv");
    if(!dictionary.is_open())
    cout << "Error : cannot read file! " << endl ;
    string line;
    while(getline(dictionary,line)){
          int x=line.find(",");
          English=line.substr(0,x);
          string temp=line.substr(x+1,line.length()-x);
          x=temp.find(",");
          French=temp.substr(0,x);
          Spanish=temp.substr(x+1,temp.length()-x);

        add(English,French,Spanish);

    }
    dictionary.close();
}
```



```
void reading2(){

    ifstream readFile2("input.txt");

     if(!readFile2.is_open())
        cout << "Error : cannot read file! " << endl ;
    string inputline2;

    while(getline(readFile2,inputline2)){
        int i=0;
        string temp=inputline2;
        cout<<"Translated Sentence in French : ";
        while(1){
        int x=temp.find(" ");

        if(x == -1){
            Word2 = temp;
            cout<< search(Word2)->FrenchWord;
            cout<<" ";
            break;
        }

        Word2=temp.substr(0,x);
        cout<<search(Word2)->FrenchWord;
        cout<<" ";
        temp=temp.substr(x+1,temp.length()-x);
        i+=x;
        }

    }
}
```

Output



C:\Users\ASUS\Music\DSA 2 Assignment\17020311\DSA_Assignment_Q2.exe

```
*******Welcome to English-French Dictionary*******

English Sentence :  I  Hometown Banuka
Translated Sentence in French : je Hometown [Banuka]

--------------------------------
Process exited with return value 3221225477
Press any key to continue . . .
```

Question 03

First, we took the string from the user input. Then asked how many substring queries they need. Then the substring queries were derived from the original string and two separate Hash functions were performed (forwardHashFunc and reverseHashFunc) to get the total hash values of the substring.

Algorithm

```
INPUT OriginalString;
INPUT number of substring queries;
INPUT relevant substrings;
DO forwardHashFunc and reverseHashFunc;
IF (forwardHash == ReversedHash)
    PRINT (is a palindrome);
ELSE
    PRINT (is not a palindrome);
```

forwardHashFunc

```
static int forwardHashFunc(String str, int len) {
        char[] chars = str.toCharArray();
        int fwdHash = 0;
        for (int i = 0; i < len; i++) {
            fwdHash += (i + 1) * (int) chars[i];
        }return (fwdHash);
    }
```

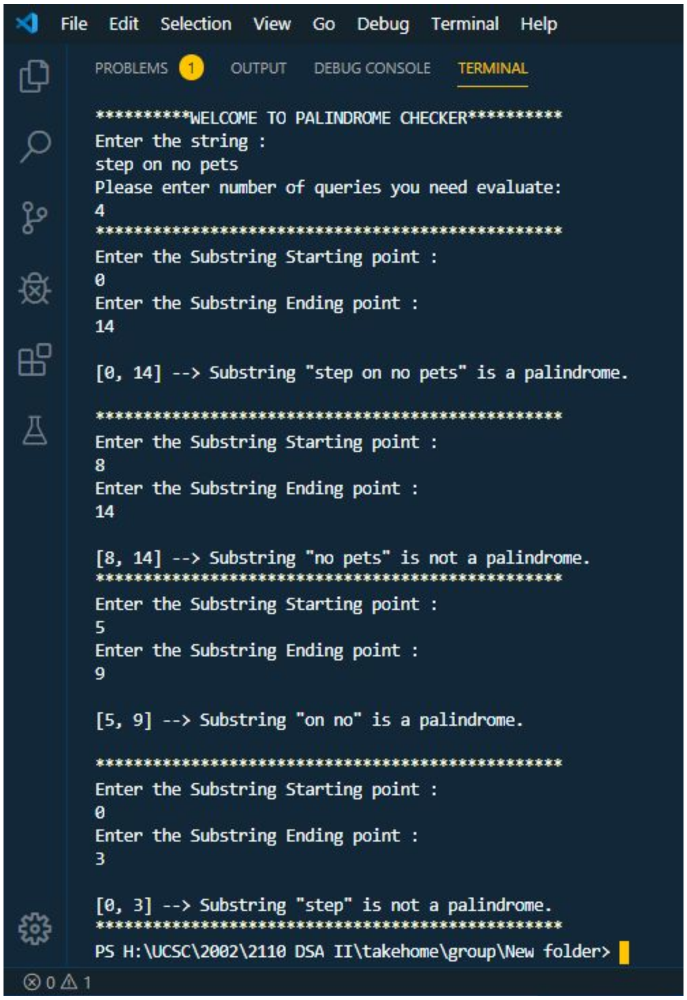
reverseHashFunc

```
static int reverseHashFunc(String str, int len) {
        String temp = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            temp = temp + str.charAt(i);
        }
        char[] chars = temp.toCharArray();
        int rvsdHash = 0;
        for (int i = 0; i < len; i++) {
            rvsdHash += (i + 1) * (int) chars[i];
        }return (rvsdHash);
    }
```

Sample Test Cases

Input String: step on no pets
Number of Queries: 4

No.	Test Case	Expected Output	Pass/Fail
1	[0, 14]	Substring “step on no pets” is a palindrome.	Pass
2	[8, 14]	Substring “no pets” is not a palindrome.	Pass
3	[5, 9]	Substring “on no” is a palindrome	Pass
4	[0, 3]	Substring “step” is not a palindrome.	Pass



```
**********WELCOME TO PALINDROME CHECKER**********
Enter the string :
step on no pets
Please enter number of queries you need evaluate:
4
**************************************************
Enter the Substring Starting point :
0
Enter the Substring Ending point :
14

[0, 14] --> Substring "step on no pets" is a palindrome.

**************************************************
Enter the Substring Starting point :
8
Enter the Substring Ending point :
14

[8, 14] --> Substring "no pets" is not a palindrome.
**************************************************
Enter the Substring Starting point :
5
Enter the Substring Ending point :
9

[5, 9] --> Substring "on no" is a palindrome.

**************************************************
Enter the Substring Starting point :
0
Enter the Substring Ending point :
3

[0, 3] --> Substring "step" is not a palindrome.
**************************************************
PS H:\UCSC\2002\2110 DSA II\takehome\group\New folder>
```