### ASSIGNMENT#2

Name: araib khan

Roll no: bcsm-f20-081

Section: 4b

1. **Which of the three original goals of the ALGOL design committee, in your opinion, was most difficult to achieve at that time?**

 Among of the three original goals of the ALGOL design committee, I think that the first goal point which is “The syntax of the language should be as close as possible to standard mathematical notation, and the programmes is available with little bit explanations. Which was very difficult at that time because it is still difficult to build a closing language because our language and computer language is not the same at first.

1. **Describe in detail the three most important reasons, in your opinion, why ALGOL 60 did not become a very widely used language.**

The three most important reasons are as under:

* The first reason is that it is not too flexible to used it is more difficult for implementation and less effective for example the pass method and the word transfer parameters in the sub-programs
* A second reason for the lack of input and outpatient statements was another major reason for its unpopularity.
* Thirdly it is not more supported by IBM no more updates features for it

1. **Was IBM’s assumption, on which it based its decision to develop PL/I, correct, given the history of computers and language developments since 1964?**

IBM, for the most part, was wrong in its view of the future of computing, at least in terms of languages. Most commercial applications are made in specific languages. So are science applications. On the other hand, IBM's 360-line computer design has been a huge success — it still dominates the computer space between computers and small computers.

1. **To understand the value of recursion in a programming language, write a program that implements quicksort, first using recursion and then without recursion**

**Without implementation:**

#include <iostream>

#include <stack>

#include <vector>

#include <algorithm>

using namespace std;

int partition(int a[], int start, int end)

{

int pivot = a[end];

int pIndex = start;

for (int i = start; i < end; i++)

{

if (a[i] <= pivot)

{

swap(a[i], a[pIndex]);

pIndex++;

}

}

swap (a[pIndex], a[end]);

return pIndex;

}

void iterativeQuicksort(int a[], int n)

{

stack<pair<int, int>> s;

int start = 0;

int end = n - 1;

s.push(make\_pair(start, end));

while (!s.empty())

{

start = s.top().first, end = s.top().second;

s.pop();

int pivot = partition(a, start, end);

if (pivot - 1 > start) {

s.push(make\_pair(start, pivot - 1));

}

if (pivot + 1 < end) {

s.push(make\_pair(pivot + 1, end));

}

}

}

**With implementation:**

#include <iostream>

using namespace std;

void swap(int\* a, int\* b)

{

    int t = \*a;

    \*a = \*b;

    \*b = t;

}

int partition (int arr[], int low, int high)

{

    int pivot = arr[high];    // pivot

    int i = (low - 1);

    for (int j = low; j <= high- 1; j++)

        if (arr[j] <= pivot)

        {

            i++;

            swap(&arr[i], &arr[j]);

        }

    }

    swap(&arr[i + 1], &arr[high]);

    return (i + 1);

}

void quickSort(int arr[], int low, int high)

{

    if (low < high)

    {

        int pivot = partition(arr, low, high);

        quickSort(arr, low, pivot - 1);

        quickSort(arr, pivot + 1, high);

    }

}

void displayArray(int arr[], int size)

{

    int i;

    for (i=0; i < size; i++)

        cout<<arr[i]<<"\t";

}

int main()

{

    int arr[] = {12,23,3,43,51,35,19,45};

    int n = sizeof(arr)/sizeof(arr[0]);

    cout<<"Input array"<<endl;

    displayArray(arr,n);

    cout<<endl;

    quickSort(arr, 0, n-1);

    cout<<"Array sorted with quick sort"<<endl;

    displayArray(arr,n);

    return 0;

}

1. **Write a grammar for the language consisting of strings that have n copies of the letter a followed by one more number of copies of the letter b, where n > 0. For example, the strings abb, aaaabbbbb, and aaaaaaaabbbbbbbbb are in the language but a, ab, ba, and aaabb are not.**

S->aSb|b

1. **Consider the following grammar: <S> → a <S> c <B> | <A> <A> → c <A> | c <B> → d Which of the following sentences are in the language generated by this grammar? a. abcd b. acccbd c. acccbcc d. accd e. ccc**

* Accd is generated
* Ccc is generated