**EXAM OF DATA STRUCTURES AND ALGORITHMS USING C++**

ACADEMIC YEAR: **2023-2024** DATE: **15/12/2023**

LEVEL: **4 (RCA YEAR 2)**

TERM: **1**

MODULE CODE: **SFEPE402**

NUMBER OF TEACHING HOURS/WEEK: **5**

DURATION: **180 Mins** MAXIMUM MARKS: **100**

**INSTRUCTIONS:**

1. Answer **ALL QUESTIONS**.
2. All resources are not allowed
3. Write your names and Group
4. Use good handwriting

**Examiners:**

HABANABASHAKA Jean Damascene

HATANGIMBABAZI Hilaire

**Questions /100 Marks**

1. Define The following terms used in DSA. **/10 Marks**
   * 1. Recursion
     2. Algorithm
     3. Algorithm Complexity
     4. Linear data structure
     5. Data structure

**Sample Answer:**

**i) Recursion** means a function is calling itself. We will use recursion whenever the solution of a problem depends on the solution of the small problem of the same nature.

ii) **Algorithm:** An **Algorithm** is a finite set of unambiguous steps or instructions to solve a given problem.

An algorithm is a **set of instructions for solving a problem** or accomplishing a task.

…Set of rules, group of steps … to solve a problem…

iii)The **complexity of an algorithm** is the amount of **Time** or **Space** required by the algorithm to process the input and produce the output.

iv)Linear Data structure: Data structure storing elements in form of a line.

**v) Data structure: Data Structure** is a logical or mathematical model for collecting and organizing data in such a way that we can perform operations on these data in an effective way.

1. What will be the output of the following C++ code?/ **2 Marks**

#include<iostream>

using namespace std;

int main(){

int a[10]={14,5,7,12,11,10,7,8,9,15};

int \*ptr = a;

cout<<\*(ptr+2)<<endl;

return 0;

}

* 1. 5
  2. 12
  3. **7**
  4. 15
  5. None of the above

**Answer: c) 7**

1. *Write symbols and explain the following Notations in Algorithms analysis.* ***6 Marks***
   1. *Omega Notation Ω*
   2. *Theta Notation Θ*
   3. *Big O Notation (****O***

***Sample Answer:***

* O (Big O) Notation: It gives the upper bound of the time/space an algorithm takes to run, in the worst-case scenario.
* Ω (Big Omega) Notation: It gives the lower bound of the time/space an algorithm takes to run, in the best-case scenario.
* Θ (Theta) Notation: It provides an average case time/space complexity.

1. Explain why Merge Sort and Quicksort are the best fastest algorithm./**2 Marks**

**Sample Answer:**

**Division and Conquer(dividing the problem into small problems)**

1. Using BIG O notation represent the following time complexity of a given algorithm 8nn+n3+6n+100  **2 Marks**

**Answer:**

**O(nn)**

1. What is the relationship between data structures and Algorithms **/4 Marks**

**Sample Answer:**

Data structures provide a framework for storing and accessing data that algorithms can operate on while

Algorithms often operate on data structures to process or manipulate data

1. Write a function named **deleteArrayElement** which delete an element in the array of size n at a given index .**/4 Marks**

Consider Array as ar[ ]={10,11,12,13,14,15};

The number of elements n is 6. Assume the index to delete is 2

The function

deleteArrayElement Would change the array to arr[]={10,11,13,14,15}

**Sample Answer:**

*#include<iostream>*

*using namespace std;*

*void deleteArrayElement(int arr[], int n,int index){*

*for(int i=index;i<n-1;i++){*

*arr[i]=arr[i+1];*

*}*

*n=n-1;*

*}*

*int main(){*

*int arr[]={52, 78, 75, 68, 88, 63, 75, 90,78};*

*int len=sizeof(arr)/sizeof(int);*

*int k=3;*

*deleteArrayElement(arr,len,k);*

*len=len-1;*

*for (int index=0; index<len-1; index++){*

*cout<< arr[index]<<endl;*

*}*

*return 0;*

*}*

1. Write using the pseudocode or natural language the following array sorting algorithms: **(20 marks)**
2. Bubble sort
3. Insertion Sort
4. Selection sort
5. Quick sort
6. Merge sort

**N.B: try to follow the pseudocode writing conventions.**

9. Express the time and space complexities in terms of Big O notation for all algorithms in question 8 respectively for input size n **(10 marks)**

10. Implement all algorithms defined in question 8 using a C++ program for each sorting algorithm. **(30 Marks)**

**///Buble sort**

11. Explain how the bubble sort algorithm is made more efficient. **(2 marks)**

**Sample Answer:**

**Use a variable check/count to check if there is no swap at a given iteration and stop there because the array is sorted.**

12.Consider the following class Square and use four(4) different techniques to create four different objects in C++. **/8 Marks**

class Square {

double side;

public:

Square(double s) {

side = s;

}

double perimeter() {

return side\*4;

}

};

**Sample Answers:**

Square \*s=new Square(8.8);

Square s1{8.8};

Square s2(8.8);

Square s3=Square(8.8);

Square s4={8.8};

Square s5{8.8};

Square s6=8.8;

**Sample Program**

#include<iostream>

using namespace std;

class Square {

double side;

public:

Square(double s) {

side = s;

}

double perimeter() {

return side\*4;

}

};

int main(){

Square \*s=new Square(8.8);

Square s1{8.8};

Square s2(8.8);

Square s3=Square(8.8);

Square s4={8.8};

Square s5{8.8};

Square s6=8.8;

cout<<s->perimeter()<<endl;

cout<<s1.perimeter()<<endl;

cout<<s2.perimeter()<<endl;

cout<<s3.perimeter()<<endl;

cout<<s4.perimeter()<<endl;

cout<<s5.perimeter()<<endl;

cout<<s6.perimeter()<<endl;

return 0;

}

**OUTPUT:**

35.2

35.2

35.2

35.2

35.2

35.2

35.2

=====================End of Exam=========================

**Merry Christmas and Happy New Year**