UNIVERSITY of SAN CARLOS
SCIENTIA • VIRTUS • DEVOTIO

DCISM
DEPARTMENT OF COMPUTER, INFORMATION SCIENCES AND MATHEMATICS

School of Arts and Sciences
Department of Computer, Information Sciences and Mathematics
CIS 2101 Final Examination
1st Semester, AY 2023-2024
December 11, 2023

Name: ____ CIS 2101 Group No. ___________

GENERAL INSTRUCTIONS. Use BLACK or BLUE ballpoint pens. Write neatly and legibly. Unreadable answers will not be corrected.

Instructions:

1) **Write your Name on the paper provide for. Put the CIS 2101 Group No. ___**
2) **Problem A: Function createInfoHeap() : FRONT of the paper starting at the left side.**
3) **Problem B: Function createEdgeList () : BACK of the paper starting at the left side.**

Programming Instructions:

1) **The return keyword will only be used if the function's return type is not void and there should 1 return statement only.**
2) **Function should not have a break or exit statement.**
3) **Proper indention should be followed and code readable.**
4) **Only efficient code gets full credit.**

Problem A: Function createInfoHeap(). [20 points]

```
typedef struct {
  char FN[24],LN[16],MI;
}nametype;

typedef struct {
  char ID[10];
  nametype name;
  char course[8];
  int yrLevel;
}studRec;

/* Definition of  student List   */
typedef struct node {
   studRec stud;
   struct node *slink;
}*studList;
```

```
#define MAX 0x2D
typedef struct {
   nametype sName;
   char ID[10];
}studInfo;

typedef struct {
   studInfo data[MAX];
   int studCnt; //elem count
}studHeap;

Function Prototype:
void swapStud
```

Function Specification:

Function createInfoHeap(). Given a linked list of student records and a course, the function will remove all student records of a given course with year level 2, extract the information needed in the heap representing the partially ordered tree and insert the record containing the information extracted into a newly created heap which will be returned to the calling function. The heap will be organized using the ID number and ID with the biggest value will be stored at the root of the POT.
Note: The following function should be called/invoked whenever necessary.

Function Prototype:

```
void swapStudInfo(studInfo *x, studInfo *y);
//Swap the 2 given studInfo
```

Problem B: Function createEdgeList() . [20 points]

```
#define SENTINEL    999      // for infinity ∞
#define MAX_VERTEX   10      //  vertices 0, 1, 2, .. 9
#define ARRAY_SIZE   0X2D

/*** Data Structure Definition ****/
typedef struct {
   int u, v;      //vertices representing the edge
   int weight;    //edge weight
}edgetype;

typedef struct node {
   edgetype edges[ARRAY_SIZE];
   int lastEdge; // index of the last existing edge in the list; -1 if empty
}*edgeList;        // Definition of the edge list

//Labeled Adjacency Matrix
typedef int adjMatrix[MAX_VERTEX][MAX_VERTEX];
```

Function Specification:

Function createEdgeList(). Given a labeled adjacency matrix representing an undirected graph, the function will create an edge list which will be returned to the calling function. It contains a list of edges arranged in ascending order according to edge weight. Note that if two or more edges have the same weight, the edge first entered into the list should retain its position. A sentinel value in the labeled adjacency matrix represents no edge. Also note that in an undirected graph the edge (u,v) = (v,u).