**COSC 2436**

**Practice Exam**

|  |  |  |  |
| --- | --- | --- | --- |
| **SECTION** | **Question** | **Value** | **Score** |
| Q1: Linked List – 35 Points | I | 8 |  |
|  | II | 8 |  |
|  | III | 3 |  |
|  | IV | 8 |  |
|  | V | 8 |  |
| Q2: Recursion – 25 Points | I | 10 |  |
|  | II | 10 |  |
|  | III | 5 |  |
| Q3:Big O – 20 Points | I | 3 |  |
|  | II | 4 |  |
|  | III | 7 |  |
|  | IV | 6 |  |
| Q4: Sorting - 20 Points | I | 5 |  |
|  | II | 10 |  |
|  | III | 5 |  |
|  | **Total** | **100** |  |

**Question 1: Linked List - 35 Points**

1. Write a Function to RECURSIVLEY count the number of nodes in a singly Linked list? This function should return the total number of nodes in the lsit

**(8 points)**

struct node {

  int val;

  node\* next;

};

int countNodes(node\* head) {

}

1. Given a double-linked list, write a function that prints nodes that are “peak”. A node is considered “peak” if the one before and after are less than the node. Desired output for given examples: (**8 points**) **Desired output:**
   1. [1,2,5,4] -> 5
   2. [1,2,6,3,7,5] -> 6,7
   3. [1,2,3] -> No output

struct node {

  int data;

  node\* prev;

  node\* next;

};

void peak(node\* head) {

}

1. **What is the output of the printList function? (3 points)**

struct Node {

  int val;

  Node\* next;

};

class LinkedList{

  private:

    Node\* head;

  public:

    LinkedList() {

      head = NULL;

    }

    void myFunc(int num) {

      Node \*myNode = new Node;

      myNode->val = num;

      if(head == NULL) {

        head = myNode;

      }

      else if(num % 2 == 0) {

        myNode->next = head;

        head = myNode;

      } else {

        Node \*curr = head;

        while(curr->next != NULL) {

          curr = curr->next;

        }

        curr->next = myNode;

      }

    }

    void printList() {

      Node\* curr = head;

      while(curr != NULL) {

        cout << curr->val << endl;

        curr = curr->next;

      }

    }

};

int main() {

  int nums[] = {1,2,4,13,10,7,9,12,6,8};

  LinkedList myList = LinkedList();

  for(int i = 0; i < 10; i++) {

    myList.myFunc(nums[i]);

  }

  myList.printList();

}

1. Given a circular linked list, write a function that prints out the next greatest value for each node. Print -1 if you’re looking at the greatest node already, meaning there is no greater node to print. Your code can print the simplified output; the extended output is just for your understanding. **(8 points)**  
   Example: input: 6,3,10,9,1

output: node 6: next greatest value = 9

node 3: next greatest value = 6

node 10: next greatest value = -1

node 9: next greatest value = 10

node 1: next greatest value = 3

simplified output: 9,6,-1,10,3

struct Node {

  int data;

  Node\* prev;

  Node\* next;

};

void nextGreatestValue(Node\* head) {

}

1. Write a function addTwoNumbers that accepts 2 doubly linked lists as input(the lists already created). It then adds the values from the nodes. Returns the head of the new list. Each node value will be an integer [0,9]. **(8 points)**

For example to add 387 + 214 = 601 will be

List 1 : 3 ⇋ 8 ⇋ 7

List 2 : 2 ⇋ 1 ⇋ 4

Result : 6 ⇋ 0 ⇋ 1

struct Node {

  int data;

  Node\* next;

};

Node\* addTwoNumbers(Node\* tail\_1, Node\* tail\_2) {

}

**Question 2: Recursion - 25 Points**

1. Write a RECURSIVE function that gets the factorial of the number n that’s passed into the function. The factorial is the product of all numbers leading up to n. For example 4! (4 factorial) = 4 \* 3 \* 2 \* 1 = 24? **(3 points)**

int factorial(int n)

{

}

1. Consider a palindrome an array of numbers, such as 76967 or 566665. Write a recursive function **palindrome** that returns true if the array is a palindrome and false if it is not

**(10 points)**

bool isPalindrome(int\* arr, int low, int high)

{

}

1. What is the output **(5 pts)**

#include<iostream.h>

void fun(int x)

{

  if(x > 1)

  {

     fun(--x);

     cout << --x <<” “;

     fun(x-1);

  }

}

int main()

{

  int a = 5;

  fun(a);

  return 0;

}

1. What is the time complexity of the following code **(3 points)**

**Question 3: Big O - 20 Points**

for(int i = 0; i < 10000000; i++) {

for(int j = 0; j < n; i++) {

      cout << "Test" << endl;

     }

}

1. What is the time complexity of the following code **(4 points)**

void myFunc(int n, int m) {

  if(n <= 0) return;

  for(int i = 0; i < m; i++) {

    myFunc(n-1,m);

  }

}

1. Give the time complexities for the following two functions and explain which one has a shorter time complexity. **(7 points)**

void func\_b(int n) {

  int i = 0;

  int j = n;

  while(i < n) {

    while(j > 0) {

      j = j - 2;

    }

    j = n;

    i += 1;

  }

void func\_a(int n) {

  int i = 0;

  int j = 2;

  while(i < n) {

    while(j < n) {

      j = j \* j;

    }

    j = 2;

    i += 1;

  }

 }

1. What is the best Time Case time complexity for Selection Sort. Describe what the list must look like for it to have this time complexity. **(6 points)**
2. Use Bubble Sort to sort the given array in ascending order. Write out the array for every iteration the sort occurs until the sorting is complete. Arr = {8,6,9,1,2,3}. DON’T Write a function just list the steps for every swap MANUALLY **(5 points)**

**Question 4: Sorting – 20 Points**

1. Given the array {24,31,12,17,9,15}. Implement a sorting function that would produce these arrays after each swap. State what type of sorting is used

**(10 points)**

Given array : {24,31,12,17,9,15}

1st swap : {9,31,12,17,24,15}

2nd swap : {9,12,31,17,24,15}

3rd swap : {9,12,15,17,24,31}

III. Use Insertion Sort to sort the given array in descending order. Write out the array for every iteration the sort occurs until the sorting is complete. Arr = {1,5,9,17,2,6}. DON’T Write a function just list the steps for every swap MANUALLY

**(5 points)**