

System Skill Midterm Quiz

Date: Saturday, October 29th, 2022

Due Date: Tuesday, November 1st, 2022 at 10.00AM

Instructor: Rachata Ausavarungnirun

Problem 1 (20 Points):

Problem 2 (30 Points):

Problem 3 (30 Points):

Problem 4 (20 Points):

Total (100 Points):

Instructions:

1. **DO NOT CHEAT.** If we catch you cheating in any shape or form, you will be penalized very heavily based on **the following plagiarism policy** ($N * 10\%$ of your total grade, where N is the number of times you plagiarized previously). This include asking the questions online, copying codes from the internet, etc.
2. Submit your work as a zip file on Canvas.
3. Because everyone has a lot of time, I expect everyone to **test your code**.
4. If not specified, input and output types are a part of the question. Please use appropriate input and output types that make sense for the purpose of the question.
5. Please clearly comment your code, especially if your code do not work perfectly.
6. Clearly indicate your final answer for each conceptual problem.
7. Your code should not have any memory leaks.

Tips:

- **Read everything.** Read all the questions on all pages first and formulate a plan.
- **Be cognizant of time.** It is a sad day if you click submit when the submission site close.
- **Canvas allows resubmission.** I will take a look at the last version you submit.
- **Show work when needed.** You will receive partial credit at the instructors' discretion.

Initials: _____

1. Linux Scripting [20 points]

Write the Linux or git scripts to handle the following tasks, please explain your answer:

- (a) You are working on the new group assignment. Let's assume that there is a starter code in on github at <https://github.com/rausavar/MUIC-s2022-t1-syskill> (Note: There is nothing there, but just assume there is a repo there). Your task is list the commands to 1) to clone the repository, 2) from the master branch that you just cloned, fork your own branch with your name as the branch name, and 3) create and then add the a new blank file called `cat` to this new branch (2.5 points)

- (b) What does the following regular expression match to: `[a-c].b*` (2.5 points)

Initials: _____

- (c) You are currently working on a series of C assignments on files `assignmentX.c`, where `X` ranges from 0 to 7. Please write a script that compiles and run all your assignments. Once all assignments are done running, the script should then send an email to yourself with the subject `Automated Results Assignments`, and the body of the email should contain all the console outputs from each assignment. Each assignment output should be separated by `==== Assignment X output =====`, where `X` is the number of that assignment. **You must use a loop and the `mail` command for this question** (15 points)

Initials: _____

2. Basic C [30 points]

- (a) In this next question, you will write a function that is used in many modern-day applications called matrix-vector multiplication that multiplies a matrix with a vector (see the definition here https://en.wikipedia.org/wiki/Matrix_multiplication).

Your task is to write a function called `int ** matrixMultiply(int **input1, int **input2, int sizeX, int sizeY, int sizeZ)` that takes in a matrix `input1` of size `sizeX` by `sizeY` and `input2` of size `sizeY` by `sizeZ`. The function should allocate the space with the correct resulting matrix size (`sizeX` by `sizeZ`) and return the pointer that point to the result of the multiplication of the two input matrices. Save this file in `matrix.c`. (25 points)

Hint: Feels free to start from our sample code that plays with matrices.

- (b) Write a main function that tests your code. Please note that you can still get a full credit on this part even if your matrix multiply does not work. (5 points)

Initials: _____

3. Linked List Redux [30 points]

Please consider the code below.

```
#include<stdio.h>
#include<stdlib.h>

struct my_node{
    int data;
    struct my_node * next;
};

typedef struct my_node Node;

Node * first; // point to the first element in my linked list

void deleteAt(int index)
{
    Node * temp;
    int count = 0;
    if(!first) return;
    else if(!index)
    {
        temp = first;
        first = temp->next;
        free(temp);
    }
    else
    {
        for(temp = first; count < index-1 && temp->next != NULL ; temp = temp->next)
            count++;
        Node * temp2 = temp->next;
        if(!temp->next) return;
        temp->next = temp->next->next;
        free(temp);
    }
}

void print()
{
    Node * current;
    for(current = first; current!=NULL; current = current->next)
    {
        printf("%d, ", current->data);
    }
}
```

Initials: _____

```
void insertAt(int index,int data)
{
    Node * temp;
    int count = 1;
    if(!(first) || index == 0)
    {
        temp = malloc(sizeof(Node));
        temp->data = data;
        temp->next = first;
        first = temp;
        return;
    }

    for(temp = first; count < index && temp->next != NULL; temp = temp->next)
        count++;
    Node * temp2;
    temp2 = malloc(sizeof(Node));
    temp2->data = data;
    temp2->next = temp->next;
    temp->next = temp2;
}

int getSize(Node * list)
{
    int count = 0;
    Node * current;
    for(current = first; current!=NULL; current = current->next)
    {
        count++;
    }
    return count;
}

int main()
{
    front = NULL;
    // Assume we do something with our list here ...
}
```

Please answer the following questions.

- (a) Is there anything wrong with the insertAt function? If so, what is wrong and please explain how to fix it. (5 points)

Initials: _____

- (b) Is there anything wrong with deletion? If so, please explain how to fix it. (5 points)

- (c) Please write the function called `int* toArray()` that returns an integer pointer to a newly allocated array containing all the items in our linked list. Your array must have the same size as the linked list. (5 points)

Initials: _____

- (d) Please write a function called `void insert(int data)`, which go through the list and inserts the data right before the node that has the data greater than `input`. For example, if your list is `{1, 4, 5, 3, 10, 6}` and you try to insert a value 7. The new list should have 7 between 3 and 10. (10 points)

Initials: _____

- (e) Please write a function called `void arrayToSortedList(int * input)`, that created a sorted link list and make `first` points to the first node in this linked list. Feels free to use any function calls we already created including `toArray` and `insert`. (5 points)

Initials: _____

4. Data in Each Bytes [20 points]

In this question, you are going to assume the following data type and its corresponding sizes: char: 1 byte and int: 5 bytes.

From a C-like snippet of a code below, answer the rest of the questions. **Note:** We use a lot of casting in the second half of this snippet to make sure things are clearly declared.

```
typedef struct type1{int i[2];} typeA;
typedef struct type2{typeA j[2];} typeB;

typeA *a;
typeB *b;
unsigned char *c;
int i;
a = malloc(sizeof(typeB));
b = a;
c = a;
for(i=0;i<15;i++)
{
    *(c+2+i) = (unsigned char)i;
}
```

Assuming that the actual location of `a` after `malloc` is called is at the physical address `0x10000` in DRAM. Using the concept of address we learn from our class, the table below shows our DRAM starting from the address `0x10000`. **Please put in the values inside each bytes in our DRAM starting at address `0x10000` up to the end of where `a` is malloced for. Write `XX` in the byte that is unknown/unassigned, and write `YY` on the byte that is not in the range that `a` covers.** I gave you the first byte, finish the rest.

Hint: Draw the organization of the struct out on a piece of paper. Pair this up with the concept of an array and type casting. This question is actually very easy but you need to be a bit careful.

Hint2: Please note that our table is actually 1D. Each row is 8 bytes in size and the address of the next row basically continues from the end of the earlier row. I just have no method to draw a very long 1D table so I need to resort to a 2D table here.

Initials: _____

Base Address in Hex (Base Address in Decimal)	Base+0	Base+1	Base+2	Base+3	Base+4	Base+5	Base+6	Base+7
0x10000 (65536)	xx							
0x10008 (65544)								
0x10010 (65552)								
0x10018 (65560)								
0x10020 (65568)								
0x10028 (65576)								
0x10030 (65584)								
0x10038 (65592)								
0x10040 (65600)								
0x10048 (65608)								