Manit Gandhi
6280004

System Skill Midterm Quiz

Date: Tuesday, October 26th, 2021
Due Date: Friday, October 29th, 2021 at 11.59PM
Instructor: Rachata Ausavarungnirun

Problem 1 (20 Points):	
Problem 2 (30 Points):	
Problem 3 (35 Points):	
Problem 4 (20 Points):	
Total (100+5 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. This is a very long exam. If you get 100, you get a full score. Any points above 100 goes to your extra credit.
3. Submit your work as a zip file on Canvas.
4. Because everyone has a lot of time, I expect everyone to **test your code**.
5. 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.
6. Please clearly comment your code, especially if your code do not work perfectly.
7. Clearly indicate your final answer for each conceptual problem.
8. 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.

1. Linux Scripting [20 points]

Provide Linux scripts to handle the following tasks:

(a) Your friend try to use the command `ls *`, it instead removes the file in your current working directory. Your friend is asking you to fix it such that he/she can just type in ls and it will list the current directory. (2.5 points)

```
unalias ls
```

(b) What does the following wildcard in shell match to: `a*b`. (2.5 points)

Any file that starts from 'a' AND ends with 'b'.

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

Any string starting with a single or more than 1 'a' and ending with 'b'.

(d) Write a regular expression that will matches any Integer. (2.5 points)

```
var r = /^[-+]?\d+$/;
```

(e) Let's say I created 5 test files called `textN.in`, where `N` is the test file number from 1 to 5, and these test files are for the determinant question from assignment 2. Write a bash script to run them all so that you can it prints all the results on the screen when the bash file is run. (10 points)

Users > manitgandhi > Desktop > Testing1 > $ code.sh

```
#!/bin/bash

# Assume the path that the user enters will contain
# testN.in files where N is 1 to 5.
# Additionally, assume that the c file det.c has been
# compiled into an executable output called 'det'

path="$1"
cd $path
for N in {1..5}
do
   echo -e '\n'
   echo -e "Test results for test$N.in \n"
   ./det < test$N.txt
   echo -e '\n'
done
```

To run this script:
./code.sh <path>

2. Basic C [30 points]

(a) Write a function called `int isPrime(int n)` that returns 1 if the number `n` is a prime number and 0 if not. Save this in `isPrime.c`. (10 points)

(b) In this next question, you will write a program that compute the dot produce of two vectors (see the definition here `https://en.wikipedia.org/wiki/Dot_product`).

Your task is to write a function called `void dotProduct(int *a, int *b, int* result, int size)` that takes in two vectors `a` and `b` of size `size`. The function should store the result in the resulting vector `result`. Save this file in `dot.c`. (20 points)

Hint: Folks, we actually have the sample code for vector add literally hosting on `syskill`. Feels free to start from there.

3. Linked List Redux [35 points]

Please consider the code below.

```
#include<stdio.h>
#include<stdlib.h>
struct node_name{
  int my_data;
  struct node_name *next;
};
typedef struct node_name Node;
Node *front;
//Delete the first occurence of my data
void delete(int data){
  Node *temp;
  Node *previous;
  if(front == NULL) return;
  else
  {
    for(temp = front; temp->my_data != data && temp->next!=NULL; temp = temp->next);
    if(temp->my_data == data)
    {
      previous->next = temp->next;
      free(temp);
    }
  }
  return;
}

//Insert a new node at the end
void insert(int data){
  Node *temp;
  Node *temp2;
  for(temp = front; temp->next != NULL; temp = temp->next);
  temp2 = malloc(sizeof(Node));
  temp2->my_data = data;
  temp2->next = NULL;
  temp->next = temp2;
}

//Just print the linked list, but make this sorted
void print(){
  Node *temp;
  printf("{");
  //TODO: IMPLEMENT THIS for part C
  printf("}\n");
}

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

Initials:

Please answer the following questions.

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

(which is required but not present here)

Yes. In insertion, a condition must be tested to check if the linked list is empty. To fix this, we check if front==NULL before the for loop. If it is empty, we will set front as the node that contains data. If it is not empty, then we will traverse and continue with the code as it is.

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

Yes. In deletion, there is no track of the previous node being kept before changing the value of temp to temp→next in the for loop. To fix this, we can add "previous=temp" before changing temp.

(c) Please write the code so that our print function can print a sorted version of our linked list to the output. (10 points)

```
//Helper function to sort

void sort (struct Node *start){
    int swapped;
    int i;
    Node * ptr1;
    Node *lptr = NULL;
    if (start == NULL)
        return;
    do {
        swapped = 0;
        ptr1 = start;
        while (ptr1->next != lptr){
            if (ptr1->data > ptr1->next->data) {
                swap (ptr1, ptr1->next);
                swapped = 1;
            }
            ptr1 = ptr1->next;
        }
        lptr = ptr1;
    }
    while (swapped);
}
```

```
// Helper function swap
void swap (struct Node *a, struct Node *b)
{
  int temp = a->data;
  a->data = b->data;
  b->data = temp;
}
```

```
//implementation part in print function
sort (temp);

while (temp != NULL)
  {
    printf("%d ", temp->data);
    temp = temp->next;
  }
```

(d) Please write a function called `void insertAt(int data, int index)`, which insert the data at the location pointed to by index. Please note that our list starts with index 0. (15 points)

```
void insertAt(int data, int index){
  int size=0;
  Node *current;
  if (index<1 || index>size+1){
      printf("Invalid index");
      exit(0);
  }
  else{
  while (index--){
    if (index==0){
      Node* newNode;
      newNode->data=data;
      newNode->next=NULL;
      Node* temp = newNode;
      temp->next=*current;
      *current=temp;
    }
    else{
      current=&(*current)->next;
    }
    size++;
  }
}
}
```

4. Where Is My Data [20 points]

In this question, you are going to assume the following data type and its corresponding sizes: `char`: 1 byte and `int`: 2 bytes.

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

```
typedef struct type1{int i[3];} typeA;
typedef struct type2{typeA j[2];} typeB;

typeA *a;
typeB *b;
unsigned char *c;
int i;
a = malloc(10000*sizeof(typeB));
b = a;
c = a;
for(i=0;i<30000;i++)
{
    *(c+i) = (unsigned char)(i%8);
}
```

typeA = 6 bytes
typeB = 12 bytes

a = 6 bytes
b = 12 bytes
c = 1 byte

Assuming that the value of `a` after `malloc` is `0x10000`. Using the concept of data representation from our class, answer the following questions.

(a) What is the size of the array `a` in bytes? (5 point)

The size of array a in bytes is 10000 x 12 = 12000 bytes

(b) What is the value of each byte of `a+2` in hex? (2.5 point)

a+2 = 0x10000 + 2.6
= 0x10000 + 12
= 0x10000 + 0xC
= 100C

100C to dec → 4108
byte 0 = 12 to hex → 0xC
byte 1 = 16 to hex → 0x10
byte 2 = 0
byte 3 = 0

(c) What is the value of `*(a+2)` in decimal number? (2.5 point)

a[2] = c[12]
c[12] = 12 % 8
12 % 8 = 4

(d) What is the value of each byte of `b+2` in hex? (2.5 point)

b+2 = 0x10000 + 2 .12
= 0x 10000 + 24
= 0x 10000 + 0x18
= 0x 10018

0x10018 to dec → 65560
byte 0 = 24 to hex → 0x12
byte 1 = 0
byte 2 = 1 to hex → 0x1
byte 3 = 0

(e) What is the value of `*(b+2)` in decimal? (2.5 point)

b[2] = c[24]
= 24%8
= 0

(f) What is the value of `c+2` in hex? (2.5 point)

c+2
= 0x10000 + 2
= 0 x10000 + 0x2
= 0x 10002

(g) What is the value of `*(c+2)` in decimal? (2.5 point)

c[2] = 2%8
= 2