CS1100 Section-1 End Sem **Date:** June 21, 2021

Duration: 1.00 hour **Max. Marks:** 20

Instructions

- This quiz contains 10 questions. All questions are compulsory.
- You need to solve this quiz on your own without referring to any source, without using internet resources and without trying codes in repl.it or any IDE. You are allowed and encouraged to work out the problems on a sheet of paper.
- Answers to this quiz have to be submitted using a google form (link shared over email). Be as precise as possible while answering the questions.
- We expect you to submit the google form only once. In case of multiple submissions, only the latest one will be considered.
- The google form will be closed at 3.05 pm. Late submissions will not be entertained.
- As a backup you are strongly encouraged to write down your answers clearly on a sheet of paper and if you face any internet issues, you can send a scanned copy of the A4 sheet to your TA. Please use this as a last resort.
- All the best!

1. (1 point) Consider the snippet of code given beside and write the output of the program. If required, you may make appropriate assumptions about the size of the basic data types.

```
void main() {
    struct point {
       int xCoord; int yCoord; int zCoord;
    };
    struct point *ptr1; char *ptr2;
    printf("diff = %ld\n", sizeof(ptr1) - sizeof(ptr2));
}
```

2. (1 point) Define a variable **ptrMat** which is a matrix of 5×3. Each element of the matrix is a pointer to an integer.

3. (1 point) The statement below can be replaced by a call to **fscanf**. Write the equivalent statement.

```
scanf ("%c", &mychar);
```

4. (1 point) Consider the following structure definitions and the declaration of variable `C`. Write down the expression which will allow you to access the xCoord of the center of the circle given by `C`. Assume that the memory allocation for all pointer variables and their initialization is taken care of appropriately in `main`.

```
struct point {
     int xCoord; int yCoord;
};
struct circle {
     struct point *center;
     int radius;
};
struct circle C;
```

5. (2 points) Study the recursive function `mystery` given beside.
   - What is the value returned by `mystery(2, 2)` ?
   - Write a closed form formula for `mystery(a, n)`.

```
int mystery(int a, int n) {
    if (a==0) return n;
    else {
       int x = mystery(a-1, n);
       return x*x;
    }
}
```

6. (2 points) What is the output of the following program? Briefly explain your answer.

```
struct classroom {
    char *className;
};
int main() {
    struct classroom C1, C2;
    C1.className = (char *) malloc (10*sizeof(char));
    strcpy (C1.className, "CS1100");
    C2 = C1;
    strcpy(C2.className, "CS2200");
    printf("%s\n",C1.className);
    printf("%s\n",C2.className);
}
```

7. (2 points) Consider an incorrect snippet of code. Assume appropriate header files are included. Correct the error(s) such that the program prints 10 on the standard output. You are not allowed to modify `main` or introduce new variables anywhere in the program.

```
1   void fun(int **ptr) {
2        ptr = (int *) malloc (sizeof (int));
3        *ptr = 10;
4   }
5   void main() {
6        int *ptr;
7        fun(&ptr);
8        printf("%d\n", *ptr);
9   }
```

8. (3 points) Study the recursive function `printer` given beside. The program prints
2 4 6 8 8 6 4 2
Determine the size of the array `A`, initialize `A` with appropriate values and fill in the blank for the call to the `printer` function in `main`.

```
void printer(int arr[], int n) {
    if (0 == n) return;
    if (n >= 2) {
        printf("%d\t", arr[0]);
        printer (arr+2, n-2);
        printf("%d\t", arr[0]);
    }
}
main() {
    int A[] = {_________________};
    // give initialization based on size of array.
    printer (A, ____);
}
```

9. (3 points) We are given an array `A` of `n` integers containing only 1s and -1s. For instance, the array `A = {1, -1, 1, -1, -1, -1, 1, -1, -1, 1}` can be an input to our program where the array size is 10. We are interested in writing a function arrange which ensures that all -1s in the array appear before the 1s. That is, after `arrange` has been invoked, `A` is arranged as: `A = {-1, -1, -1, -1, -1, -1, 1, 1, 1, 1}`. Note that the number of 1s and -1s in the array can be arbitrary. Fill in the blanks to complete the function.
The function swap is provided which needs to be invoked from arrange with appropriate parameters.

```
void swap(int *p, int *q) {
     int temp = *p; *p = *q; *q = temp;
}
void arrange(int A[], int n) {
     int left = 0; int right = n-1;
     int count = 0;
     while (count != n) {
         switch (A[left]) {
              case 1:  swap(_____, _____);
                       right--; break;
              case -1:  ______;  break;
         }
         count++;
     }
}
```

10. (4 points) In this question you are given a **non-empty** singly linked list with **positive integer data** item. You need to write a **recursive function** to return two values, the maximum of the odd integers and the maximum of the even integers in the linked list. For example, if the linked list contains data $\{24, 3, 6, 9, 18, 12, 15, 21\}$ then the maximum of odd integers is 21 whereas the maximum of the even integers is 24. You can use the function max which is used to find the maximum of two integers.

    Fill in the blanks to complete the recursive function `findMax`. The `main` function is given for you to understand how findMax is invoked. You do not have to write anything in `main`.

```
struct node {
    int data;
    struct node *next;
};
int max(int a, int b){
    if (a > b) return a;
    else return b;
}
void main() {
   struct node *head;
   int maxOdd; int maxEven;
   // code to populate linked list.
   findMax(head, &maxOdd, &maxEven);
   printf("Max of odd values = %d\n", maxOdd);
   printf("Max of even values = %d\n", maxEven);
}
```

```
void findMax(struct node* curr, int* oddMax, int* evenMax) {
        if (curr == NULL) { _______;  _______;  return;}
        else {
               findMax(curr->next, oddMax, evenMax);
               if (curr->data % 2 == 0)  ___________;
               else  ___________;
        }
}
```

This is the end of the Question Paper.
Hope you enjoyed it.
