COMP 2510 Midterm Exam Winter 2023

This 90-minute exam has 6 questions. Scan the whole test before starting. Budget your time wisely. You may NOT tear the pages apart.

It is a violation of the Honour Code to look at any exam other than your own, to look at any other reference material, or to otherwise give or receive unauthorized help. Violation of the Honour Code will result in ZERO.

You are free to use blank sheets attached at the end of the exam as your work area. However, only answers in the answer box will be graded. If you make any assumption, make sure to write them in the answer box.

You will be expected to write C code on this exam. We recommend that you properly indent your code. Beyond that, you may use any C feature that you have learned about in class.

You have an option to choose Q4 or Q5. You only need to do one question. Doing additional question will not earn you additional points. Please write down your selected question between Q4 and Q5: Q4

Q0	1
Q1	3
Q2	5
Q3	1
Q4	10
Q5	—
Total	20

0. [1 points] Write last name, first name and A number at the top of each page (front and back) including unused pages and scratch paper.

1. [3 points] Following instructions on the front page, in question descriptions and announcement during the exam. Not following any instruction will result in 0 for this question.

2. [5 points] Complete the program below so that it prints out an Ascii "box" of size n, with a forwards slash, where n is the integer entered by the user. As an example, here is the corresponding box for size n = 5:

```
*****
*  **
* * *
**  *
*****
```

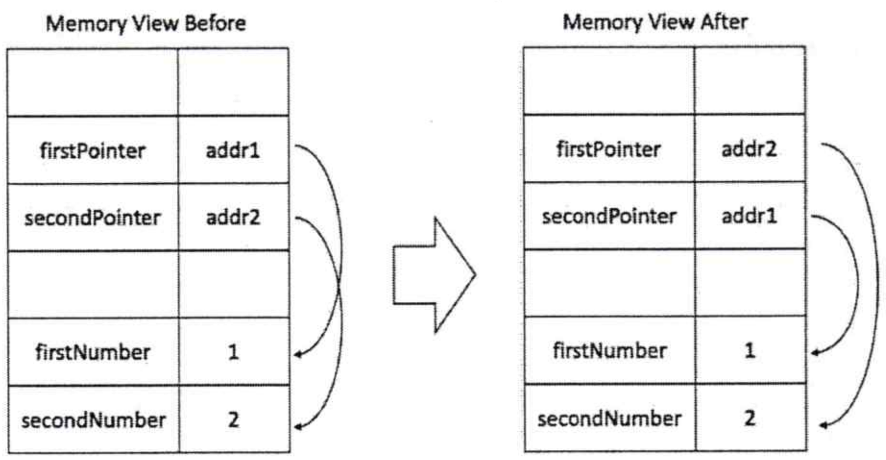
For your solution, no loops are allowed. Using a loop will result in 0 for this problem.

```
int main() {
   int n;
   printf("What is the size of your box w/for. slash?\n");
   scanf("%d", &n);

   print_box(n, n);

   return 0;
}

void print_box( int m , int n        ){
   if (m==1 && n==1) {
     return 0;
   }
   if( m*n %5==0 ){
     printf("\n");
   }
   if( m==0 || n==0 ){
     return 0;
   }
   if(m*n =16 || m*n ==12 ||
     m*n = 6 ||(m*n ==4 && m != n)) {
     printf(' ');
   }
```

→ next page

```
        printf("*");
        print_box(m, n-1);
        print_box(m-1, n);
}
```

3. [5 points] Write a function that swaps the contents of pointers. The pictorial illustration of what your function needs to achieve is shown below. Make sure to write any missing information in the code snippet below. Your solution in swapPointers should not be more than 5 lines of code.



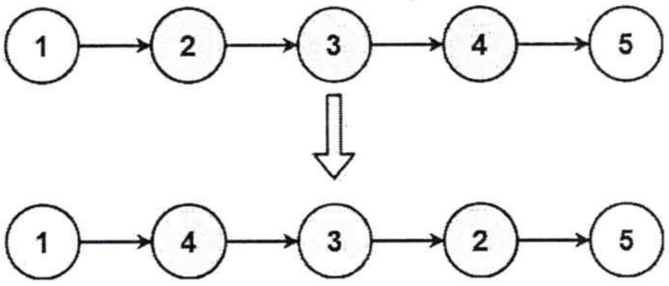
```
void swapPointers(int* firstPointer, int* secondPointer
// No more than 5 lines of code here
   int* tmp;
   tmp = firstPointer;
   firstPointer = secondPointer;
   secondPointer = tmp;


}

int main(){
      int firstNumber = 1;
      int secondNumber = 2;

      int *firstPointer = &firstNumber;

      int *secondPointer = &secondNumber;

      swapPointers(*firstPointer, *secondPointer );

      return 0
}
```

-4

4. [10 points] In poker, a straight is a hand that contains all five cards of consecutive numbers. Write a function that takes in a pointer to an array hand and returns 1 if the hand the pointer points to is a straight and 0 otherwise. The prototype is provided for you below. Feel free to add a helper function below isStraight if you'd like. Note that char *hand contains valid card hands only such as A, 2, 3,…, 9, T, J, Q, K (note that 10 is represented by T). Therefore, A, 2, 3, 4, 5 is a valid straight as well as 9, T, J, Q, K. The input is not sorted, so 5, 3, 2, 6, 4 is a valid straight. There is a wraparound, which means J, Q, K, A, 2 is a straight. No corner case testing is required and you are allowed to modify the input char.

```
~~int isStraight(char *hand) {~~
~~}~~
int main() {
   int n= 17;
   char origin[17]={'A', '2', '3', '4', '5', '6', '7', '8', '9', 'T', 'J', 'Q', 'K', 'A', '2', '3', '4'}
   int i=0, j=0;
   boolean ~~result~~ = 0;
   char hand={'A', '2', '3', '4', '5'}
   while(i<n){
      if(origin[i]==hand[j]){
         origin[i]='0';
         i++;
         j++;
      }
   }
   for(int k=4; k<n; k++){
      if(origin[k]=='0' && origin[k-1]=='0'&&origin [k-2]=='0'&&
         origin [k-3] =='0'&& orig[k-4]=='0'){
         result=1;
      }
   }
   return result;
}
```

5. [10 points] Given the head of a singly linked list and two integers left and right where left <= right, reverse the nodes of the list from position left to position right, and return the head of the reversed list. Your solution must be in O(N) time complexity and O(1) space complexity.



Input: head = [1,2,3,4,5], left = 2, right = 4
Output: [1,4,3,2,5]

```
typedef struct ListNode{
     int data;
     struct ListNode* next;
}ListNode;

ListNode* reverseBetween(ListNode* head, int m, int n) {
```