Toic: pointer, structure and link list.

Pointer:

Question 1: Complete the function void update(int *a,int *b). It receives two integer pointers, int* a and int* b. Set the value of a to their sum, and b to their absolute difference. There is no return value, and no return statement is needed. [Add any libraries you want]

a’=a+b
b’=|a-b|

```
#include <stdio.h>
void update(int *a,int *b) {
   // Complete this function
}
int main() {
   int a, b;
   int *pa = &a, *pb = &b;

   scanf("%d %d", &a, &b);
   update(pa, pb);
   printf("%d\n%d", a, b);

   return 0;
}
```

Question 2: Implement Call by reference using pointers to add two numbers.

Question 3: Write a program in C to print all permutations of a given string using pointers.

Question 4: Write a program in C to print all the alphabets using a pointer.

Question 5: Write a C program to add two matrices using pointers. C program to input two matrices from the user and find the sum of both matrices using pointers.

Structures and Unions

Question 1: Write a program in C to show the usage of pointer to structure.

Question 2: Write a program in C to show a pointer to union.

Question 3: What is output of the following C code:

```
#include<stdio.h>
struct st
{
    int x;
    struct st next;
};

int main()
{
    struct st temp;
    temp.x = 10;
    temp.next = temp;
    printf("%d", temp.next.x);
    return 0;
}
```

(A) Compiler Error
(B) 10
(C) Runtime Error
(D) Garbage Value

Linked List:

Q1:What is the output of the following function for starting pointing to the first node of the following linked list? **1->2->3->4->5->6**

```
void fun(struct node* start) {
        if(start == NULL)
                return;

        printf("%d ", start->data);

        if(start->next != NULL )
                fun(start->next->next);

        printf("%d ", start->data);
}
```

A. 1 4 6 6 4 1
B. 1 3 5 1 3 5
C. 1 2 3 5
D. 1 3 5 5 3 1

Q2. What does the following function do for a given Linked List with the first node as head?

```
void fun1(struct node* head) {
        if(head == NULL)
                return;
        fun1(head->next);
        printf("%d ", head->data);
}
```

A. Prints all nodes of linked lists

B. Prints all nodes of linked list in reverse order

C. Prints alternate nodes of Linked List

D. Prints alternate nodes in reverse order

Q3: Consider an implementation of an unsorted singly linked list.

Suppose it has its representation with a head pointer only.

Given the representation, which of the following operations can be implemented in O(1) time?

i) Insertion at the front of the linked list

ii) Insertion at the end of the linked list

iii) Deletion of the front node of the linked list

iv) Deletion of the last node of the linked list

A. I and II

B. I and III

C. I, II and III

D. I, II and IV

Q4: Give a Θ(n)-time nonrecursive procedure that reverses a singly linked list of n elements. The procedure should use no more than constant storage beyond that needed for the list itself

Q5: Which of the following sorting algorithms can be used to sort a

random linked list with minimum time complexity?

A. Insertion Sort

B. Quick Sort

C. Heap Sort

D. Merge Sort