Quiz – 1

Data Structures (CSD102)

Date: 17-9-2024 [**Total Marks = 15]** **Time: 4:00pm – 4:45pm**

1. Consider the following function that takes reference to head of a Doubly Linked List as a parameter. Assume that a node of a doubly linked list has the previous pointer as prev and next pointer as next.

```
void fun(struct node **head_ref)
{
  struct node *temp = NULL;
  struct node *current = *head_ref;

  while (current !=  NULL)
  {
    temp = current->prev;
    current->prev = current->next;
    current->next = temp;
    current = current->prev;
  }

  if(temp != NULL )
    *head_ref = temp->prev;
}
```

Assume that reference of the head of following doubly linked list is passed to above function 1 <--> 2 <--> 3 <--> 4 <--> 5 <-->6. What should be the modified linked list after the function call? **[2 Marks]**

Answer:

6 <--> 5 <--> 4 <--> 3 <--> 2 <-->1

2.

What is the output of the following C-program?

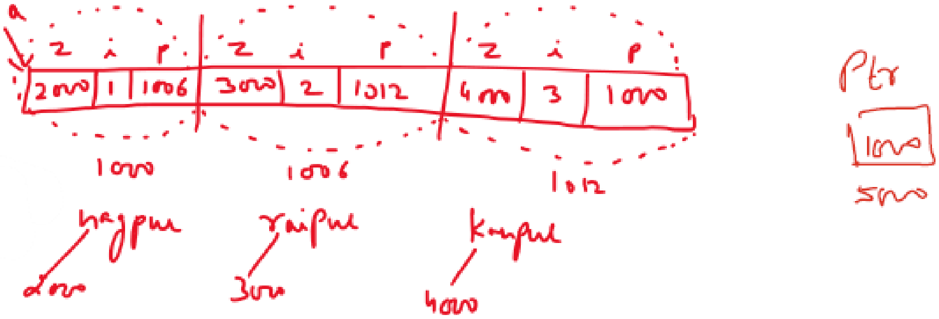
```
#include<stdio.h>
main()
{
        struct s1
        {
                char *z;        => 2B
                int i;          => 2B
                struct s1 *p;   => 2B
        };                                    6B
        struct s1 a[] = {{"nagpur", 1, a+1},        2000  3000
                         {"raipur", 2, a+2},        4000
                         {"kanpur", 3, a}};

        struct s1 *ptr = a;

        printf("%s%s%s", a[0].z, ptr->z, a[2].p->z);
                         1000    1000     1012
                         2000    2000     1000
                                          2000
}
```

nagpur, nagpur, nagpur

[3 Marks]



3. What will be the increasing order of asymptotic complexity for the following functions?

$F_1(n) = 2^n$ $F_2(n) = n^{3/2}$

$F_3(n) = n\log n$ $F_4(n) = n^{\log n}$ **[2 Marks]**

$f_3 \; f_2 \; f_4 \; f_1$

Let $n = 1024$

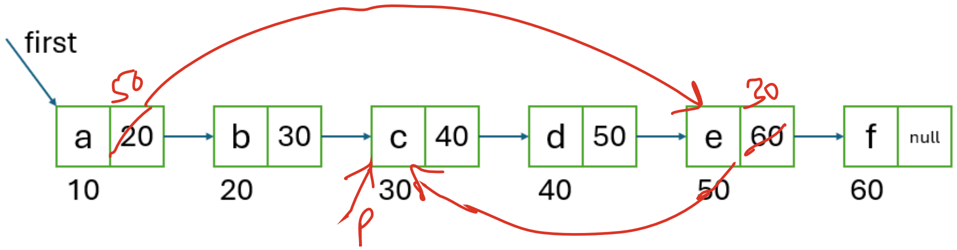
$F_1 = 2^{1024}$

$f_2 = (1024)^{3/2} = (2^{10})^{3/2} = 2^{15}$

$f_3 = (1024)\log(1024) = 2^{10}\log 2^{10} = 10\times 2^{10}\overset{1}{\log_2 2} = 10\times 2^{10}$

$f_4 = (1024)^{\log(1024)} = 2^{10\log_2 2^{10}} = 2^{100\log_2 2} = 2^{100}$

4. Consider the following linked list



What could be the output after executing the following sequence of statement?

I. struct node *p;

II. p = first->next->next;

III. first->next = p->next->next; **[1 Mark]**

IV. p->next->next->next = p; **[1 Mark]**

V. printf(first->next->next->next->next->data); e **[2 Marks]**

5.

lb_1, ub_1, lb_2, ub_2, c

A 2-D array A[-4…7, -1…3] requires 2 bytes of storage space for each element. If the array is stored in row-major form having base address 500, then the address of A[6, 2] will be?

BA $\quad i \quad j$

$$RMO = BA + [(i - lb_1) * nc + (j - lb_2)] * c$$

$$nc = ub_2 - lb_2 + 1$$

$$nc = 3+1+1$$

$$A[6,2] = 500 + [(6+4)*5 + (2+1)]*2$$

$$= 500 + [50+3]*2 = \underline{606}$$

[Row-major formula (1 Mark) + address (2 Mark)]

6. While applying insertion sort to insert every element we are using linear search to find the correct place of that element then what will be the time complexity of insertion sort in the worst case if you use binary search in the place of linear search? Justify your answer.

[2 Marks]

Answer:

	With linear search		**With binary search**	
Number of elements, n	Comparisons	Swaps	Comparisons	Swaps
n = 1	0	0	0	0
n = 2	1	1	log 2	1
n = 3	2	2	log 3	2
.	.	.	.	.
.	.	.	.	.
n = n	n-1	n-1	log n	n-1
Sum of n natural number	0+1+2+…n-1 = n^2	n^2	n log n	n^2

Time complexity of insertion sort in the worst case while using binary search = nlogn + n^2 => $O(n^2)$ [2 marks]