

Subhojit Ghimire - 1912160

Name: Subhojit Ghimire

Sch. Id.: 1912160

Subject: Data Structures

Subject Code: CS 201

Semester: IIIrd

Date: 19th October, 2020

UG Mid Sem Examination, 2020

Branch: CSE

Q

Ans: Given function;

```
void foo(int m)
```

```
{  
    for (int j=1; j<=m; ++j)
```

```
{  
        k*=j; // k = k*j
```

```
}  
}
```

```
for (int i=0; i<=foo(n); ++i)
```

Here

~~foo(n) is O(n)~~, so,

foo(n) ; so, $O(n)$

foo(int m) is in void,

so, $O(1)$

\therefore The time complexity = $O(n) + O(1)$
= $O(n)$

Q.20 (a) Ans:

Some of the advantages of using linked lists over arrays are:

- (i) linked list elements can be easily inserted or removed without reallocation of the entire structure.
- (ii) Linked lists are dynamic, i.e., the length of list can be increased or decreased as required.

Q.20 (b) Ans)

The data types in C can be categorised in few types and all of them have different data type specifiers.

1. Basic or Main type: Examples: (i) int
(ii) char
(iii) float
(iv) double.

All these are present in numerous combinations of signed, unsigned, long and short.

2. Void Type: Examples: (i) void exit(int);
(ii) int abc(void);
(iii) void * malloc(size);

3. Boolean Type: Example: (i) _Bool variable;

4. Enumeration Type: Example: (i) enum

5. Derived Type: Example: (i) Pointertype
(ii) Array type
(iii) Structure type
(iv) Union type
(v) function type.

Q.30

Ans

```
struct NODE
{
    int data-value;
    struct NODE * before, * after;
};

void Pop ( struct NODE *given-node)
{
    given-node->before = (given-node->before)->before;
    given-node->after = (given-node->after)->after;
}
```

Discussing necessary conditions, in sentences;

(i) if, ~~either~~^{both} given-node → before == NULL
and given-node → after == NULL

Then, those locations do not exist and hence,
only the given node exists,

Hence, printf("UNDERFLOW!");

(ii) if, given-node → before == NULL,
printf("PREVIOUS NODE DOES NOT EXIST");

if given-node → after == NULL,
printf("NEXT NODE DOES NOT EXIST");

(iii) if (given-node → before) → before == NULL
given-node → before = NULL;

if (given-node → after) → after == NULL
given-node → after = NULL;

Q.4

Ans

Given, 1 Million integer data in RAM
1 Trillion integer data in HDD
Method of Sorting: External Sorting.

The large elements cannot be checked and validated due to large size, so array type is used and data is divided and checked using Merge Sorting technique.

Algorithm to validate whether the given data of HDD are sorted in ascending order or not:

(i) Read in-file such that most num-size elements can be read at a time.

(ii) Merge Sort the taken elements (works for upto 10^6 data).

(iii) Validate data in array "i" for

(iv) for $10^6 \times 10^6 = 10^{12}$, check each element.

(v) If, $i^{th} > (i+1)^{th}$, print "NOT SORTED"

Else print "SORTED"

Program function:

```
void check (long long inp[!], long long n)
{
    int flag = 1;

    for (long long i = 0 ; i < n-1 ; ++i)
    {
        if (inp[i] > inp[i+1])
        {
            flag = 0;
            break;
        }
    }

    if (flag)
        printf("SORTED");
    else
        printf("NOT SORTED");
}
```