

## Assignment - 4

Pocharan Sai

API9110010032

1) Write a programme to insert and delete an element at the  $n^{\text{th}}$  and  $k^{\text{th}}$  pointer in a linked list where  $n$  and  $k$  are taken from the user.

```
A) #include <stdio.h>
#include <stdlib.h>

struct node {
    int data;
    struct node *next;
};

struct node *head;

void Insert (int data, int n) {
    Node *temp = new node();
    temp->data = data;
    temp->next = NULL;
    if (n == 1) {
        temp->next = head;
        head = temp;
    }
    return;
}
```

```

void Delete - (int k) {
    struct Node * temp = head;
    if (k == 1) {
        head = temp->next;
        free(temp);
        return;
    }

```

```

    Node * temp = head;
    for (int i = 0; i < k - 2; i++) {
        temp = temp->next;
    }
    temp->next = temp->next->next;
    temp->next = temp;
}
void print();

```

```

for (int i = 0; i < k - 2; i++)
    temp = temp->next;
    free(temp);
}

```

```

int main() {
    int n, x, k;
    head = NULL;
    printf ("Enter the position for and deleting
            and inserting; ");
    scanf ("%d", &n);
    scanf ("%d", &x);
    Insert (n, x);
    printf ("Enter the position to delete);
    scanf ("%d", &k);
    delete (k);
    Print (x);
    return;
}

```

Q) Construct a new linked list by merging alternative nodes and two lists for example in list 1 we have { 1, 2, 3 } and list 2 <sup>4, 5, 6</sup> and in the new we should have { 1, 4, 2, 5, 3, 6 }

A) #include <stdio.h>

#include <stdlib.h>

struct node {

int data;

struct node \*next;

}

void print list (struct node \*head)

{

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

ptr = ptr->next;

printf ("Null/n");

}

push

void push (struct node \*head, int data)

{

struct node \*new = (struct node \*) malloc  
(size of (struct node));

new->data = data;

new->next = \*head;

\*head = new;

}



```
struct node * merge (struct node * a, struct  
node * b)
```

```
{ struct node fake;
```

```
struct node * tail = fake;
```

```
fake->next = null;
```

```
while(1)
```

```
{ if (a == null)
```

```
{ tail->next = b;
```

```
break;
```

```
}
```

```
else if (b == null)
```

```
{ tail->next = a;
```

```
break;
```

```
}
```

```
else
```

```
{
```

```
tail->next = a;
```

```
tail = a;
```

```
a = a->next;
```

```
tail->next = b;
```

```
}
```

```
} return fake->next;
```

```

}
void main()
{
    int keys[] = {1, 2, 3, 4, 5, 6, 7};
    int n = size of (keys) / size of key[0];
    struct node *a = NULL; *b = NULL;
    for (int i = n-1; i > 0; i = i-1)
        push(&a, keys[i]);
    for (int i = n-2; i >= 0; i = i-2)
        push(&b, keys[i]);
    struct node *head = merge(a, b);
    printlist(head);
}

```

3. Find all the elements in the stack whose sum is equal to k

A) #include <stdio.h>

```

void find(int arr[], int a, int k) {
    int total = 0;
    int x = 0, y = 0;

```

```

for (x=0; x<a; x++) {
    while (sum<k && y<a)
        total = arr[y];
        y++;
}

```

```

for (x=0; x<a; x++) {
    while (total<k && y<a)
        total = arr[y];
        y++;
    if (total == 0)
        {
            printf("find");
            return 0;
        }
}

```

```

    total -= arr[x];
}
int main(void) {
    int arr[] = {9, 10, 12, 4, 1, 2, 3};
    int k = 565;
    int a = size of (arr) / size of (arr[0]);
    find(arr, a, k);
    return 0;
}

```

4) write a programme to print elements of Queue?

i) Reverse Order ii) Alternate Order.

```
#include <stdio.h>
```

```
#define size 20
```

```
void insert(int);
```

```
void delete();
```

```
int queue[size], a = -1, b = -1;
```

```
void main()
```

```
{ int num, choice;
```

```
while (1)
```

```
{ printf("\n New\n");
```

```
printf("1. Insert\n2. Delete\n3. Print
```

```
4. Reverse\n5. Alternate\n6. Exit);
```

```
printf("Enter your choice. ");
```

```
scanf("%d", &choice);
```

```
switch(choice)
```

```
{ case 1: printf("Enter the num to insert");
```

```
scanf("%d", &numchoice);
```

```
insert(num);
```



break;

case 2:

```
printf("Reverse queue");  
for(int i=size, i>0; i--)  
{  
    if(queue[i]==0)  
        continue;  
    printf("%d", queue[i]);  
}  
break;
```

case 3:

```
printf("Alternate elements");  
for(int i=0, i<size, i>0, i+=2)  
{  
    if(queue[i]==0)  
        continue;  
    printf("%d", queue[i]);  
}  
break;
```

return 0;

}

1) i) How array is different from linked list?

2) write a programme to add first element of one list to another list for example we have (1,2,3) in list 1 and (4,5,6) in list 2 we have to get (4,1,2,3) as output for list 1, and (5,6) list 2.

i) Arrays vs linked lists

1. Both are the data structures. Both are used to store the data.

2. Cost of accessing the elements

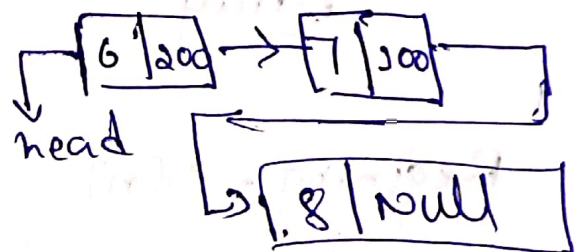
Arrays



⇒ it takes at constant time

$$O(1)$$

linked list



⇒ it depends on number of nodes in the linked list

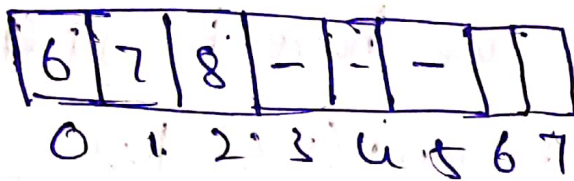
$$O(n)$$

### 3. Memory Requirement and Utilization

#### Array

⇒ Ineffective in memory utilization

Ex)



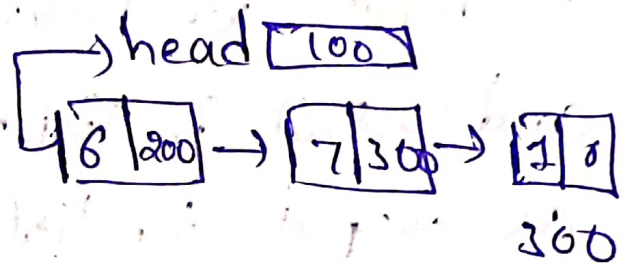
$$8 \times 4 = 32 \text{ bytes}$$

$$\text{Used} = 12$$

⇒ Require memory in less

#### Linked List

⇒ it is in dynamic size



$$8 \times 3 = 24 \text{ bytes}$$

⇒ More requirement

### 4. Cost of insertion and cost of deletion

#### Array

Beginning -  $O(n)$

At end -  $O(1)$

$i$ th position -  $O(n)$

#### Linked List

$O(1)$

$O(n)$

$O(n)$

## 5. Easy use and Operations:

Array

linked lists

=> easier to use

=> less easier

=> linear and  
Binary

=> linear

ii) # include <stdio.h>

# include <stdlib.h>

int len(int a[])

{  
int i = 0, xy = 0;

while (1)

{  
if (a[i] != 0)

{  
xy++, i++;

}  
else

{  
break;

}  
}  
return xy;



2

```
void change list (int a[], int q[])
```

```
{ for (int i = len(a) - 1; i >= 0; i--)
```

```
{ a[i+1] = a[i];
```

```
}
```

```
a[0] = a[0];
```

```
printf ("n Elements of old array: \n")
```

```
for (int i = 0; i < len(a); i++)
```

```
{ printf ("%d", a[i]);
```

```
}
```

```
for (int i = 0; i < len(y); i++)
```

```
{ y[i] = y[i+1]; }
```

```
printf ("n elements of new array: \n")
```

```
for (int i = 0; i < len(a) ; i++)
```

```
{ print f("%d", a[i]);
```

```
}  
int main()
```

```
{ int x[10] = {1, 2, 3}, a[10] = {4, 5, 6};
```

```
    change list = (a, b);
```

```
}
```