FIRST SEMESTER MCA (2020 SCHEME)

Practical Examination June 2021

20 MCA185 DATA STRUCTURES LAB

Date: 30-6-2021 Time: 9:30 am - 12:30 pm.

Submitted by
Anjana.V
ICE20MCA2012

1. merging of two arrays

Algorithm

Step 1: Start
Step 2: declare 3 integer array and variable m, n, i, j, k=0;
Step 3: Read the no. of elements in Host array as m
and Read the elements in Host array,
Step 4: Read the no. of elements in Second array as n
and read the elements in Second array;
Step 5: Initialise i and j as 0.
Step 6: repeat Step 7 to Step 10. until i<m && j<n
Step 7: check if a[i] < b[j] then.
Step 8: assign the element a[i] to c[k] and increment i by 1.
Step 9: else assign the element b[j] to c[k]
Step 10: Increment j by 1 and k by 1
Step 11: check i >= m then.
Step 12: repeat Step 13 to Step Step 14 until j<n.
Step 13: assign the element in a and b[j] to c[k]
Step 14: Increment i and k by 1.

Step15: Check $j \geq n$ then

Step16: Repeat Step 17 to Step18 until $j < m$

Step17: Assign the element in $a[i]$ to $c[k]$.

Step18: Increment j and k by 1

Step19: Print the merged array.

Step20: Stop.

Default input and output

Input

Element First array 1 2 3

Element in second array 5 6 7 9

output

merged array : 1 2 3 5 6 7 9

Program code

```
# include <stdio.h>
# include <conio.h>
  Void main()
{ int arr1[50], arr2[50], arr3[100], m, n, j, i, k=0;
  Printf (" Enter the number of element in 1st array :\n");
    Scanf ("%d", &m);
  Printf ( " Enter the number of elements:\n");
      For(i=0; i<m; i++)
        S canf( "%d", arr1 [i]);
  Printf( " Enter the no. of element in II nd array :\n");
     Scanf ( " %d", &n);
   Printf ( " enter the elements");
   For (i=0; i<n; i++)
```

```
Scanf("%d", &arr2[i]);
  i=0; j=0;
while (i<m && j<n)
 { if(arr1[i]<arr2 [i])
   {
     arr3 [k] = arr1 [i];
       i++;
    }
   else
      arr3[k] = arr [j];
          j++
   }
     k++;
  {
    if (i>=m)
    while (j<n)
   [ arr3[k] =arr[j];
       j++; k++;
   }
}
IF (j>=m)
{ while (i<m)
  { arr3 [k] =arr1[i];
      i++; j++;
  }
  } Printf("merged array :\n");
    for (i=0; i<m; i++)
    { Printf( "%d", arr3[i]);
    }
     getch ();
}
```

2. implement circular queue.

Algorithm

Step1: Start
Step2: define max as 5
Step3: declare an integer array of size max and few variable as front and rear initialize them with -1
Step3: declare two integer variables choice item.
Step4: repeat step 4 to step 11. until choice 4.
Step5: Print the menu and read choice as
Step6: repeat step 7 to step11 until if choice is 1
Step7: read the value as item.
Step8: call the insert function.
Step9: if choice is 2 call deletion function.
Step10: if choice is 3 call display function
Step11: if choice is 4 exit
Step12: stop.

Insert function

Step1: Start
Step2: check front=0 and rear = max-1 then.
Step3: print queue overflow and return.
Step4: check if front = -1 then,
Step5: Set front and rear as 0
Step6: else, check if rear = max-1 then,
Step7: Set rear as 0
Step8: else, increment rear by 1
Step9: assign item to the array

Deletion function

Step 1 : Start

Step 2 : Check if front = -1 then,

Step 3 : Print queue overflow and return.

Step 4 : Print the element deleted from the queue

Step 5 : Check if (front == rear) then,

Step 6 : front and rear as -1

Step 7 : else, check if front = max-1 then

Step 8 : Assign front as 0

Step 9 : else increment front by 1

Step 10 : Stop

display function

Step 1 : Start

Step 2 : Initialize front - pos = front and rear - pos = rear

Step 3 : Check if front = -1 then,

Step 4 : Print queue is empty and return

Step 5 : Print the queue elements

Step 6 : Check if front - pos < = rear - pos

Step 7 : repeat Step 8 to Step 8 until front - pos < = rear - pos.

Step 8 : Print the value of front - pos and increment front pos by 1

Step 9 : else repeat step 10 to until front - pos < = max - 1

Step 10 : Print front pos in array and increment front - pos by 1

Step 11 : Set front pos as 0

Step 12 : Repeat Step 13 to Step 14 until front pos ≤ rear - pos

Step 13: Print Value of front in array

Step 14: Increment front-pos by 1

Step 15: Stop.

Default Input and output

Choice: 1

element: 23

Choice: 1

element: 25

Choice: 2

deleted element is 23

Choice 3

Queue element are 25.

Program code

```
#include <stdio.h>
#include <conio.h>
#define MAX 5
int Queue_arr[MAX];
int front = -1; int rear = -1;
void insert (int item)
{
    if ((front == 0 && rear == MAX-1) || (front == front+1))
    {
        Printf ("Queue overflow");
        return;
    }
    if (front == -1)
    {
        front = 0, rear = 0;
```

```
else
  if (rear == MAX-1)
     rear=0;
  else
     rear=rear+1;
}
C queue_arr [rear] = item;
}
  void deletion()
  {
     if (front == -1)
     {
        Print ("Queue underflow");
        return;
     }
    Printf(" element deleted from Queue is i%.d \n", Cqueue_arr[front])
       if (front == rear)
       { front = -1, rear = -1;
       }
       else
       {
         if (front = MAX-1)
           front = 0;
         else
           front = front + 1;
       }
  }
  void display()
  {
     int front_pos = front, rear_pos = rear;
        if (front == -1)
        {
          Print ("Queue is empty\n");
             return;
        }
        Print ("Queue element \n");
        if (front_pos <= rear_pos)
        While (front_pos <= rear_pos)
     { Print ("%d", Cqueue_arr [front_pos]);
        front_pos++;
     }
```

```
else
{ while (front - pos <= rear-1)
  }
  Printf("%d", Queue_arr[front-pos]);
    front-pos++;
  }
}
void main();
{ int choice, item;
  do
  { Printf(" 1. Insertion \n 2. deletion \n 3. Display \n 4. Exit \n")
    Printf("Enter your choice : \n");
     Scanf("%d", &choice);
    Switch (choice);
  { case 1 : Printf(" Input element for insertion ");
             Scanf ("%d", item); insert (item);
                  break;
    case 2 : deletion (); break;
    case 3 : display (); break;
    case 4 : break;
  }
  }
  while (choice != 4);
     getch();
}
```