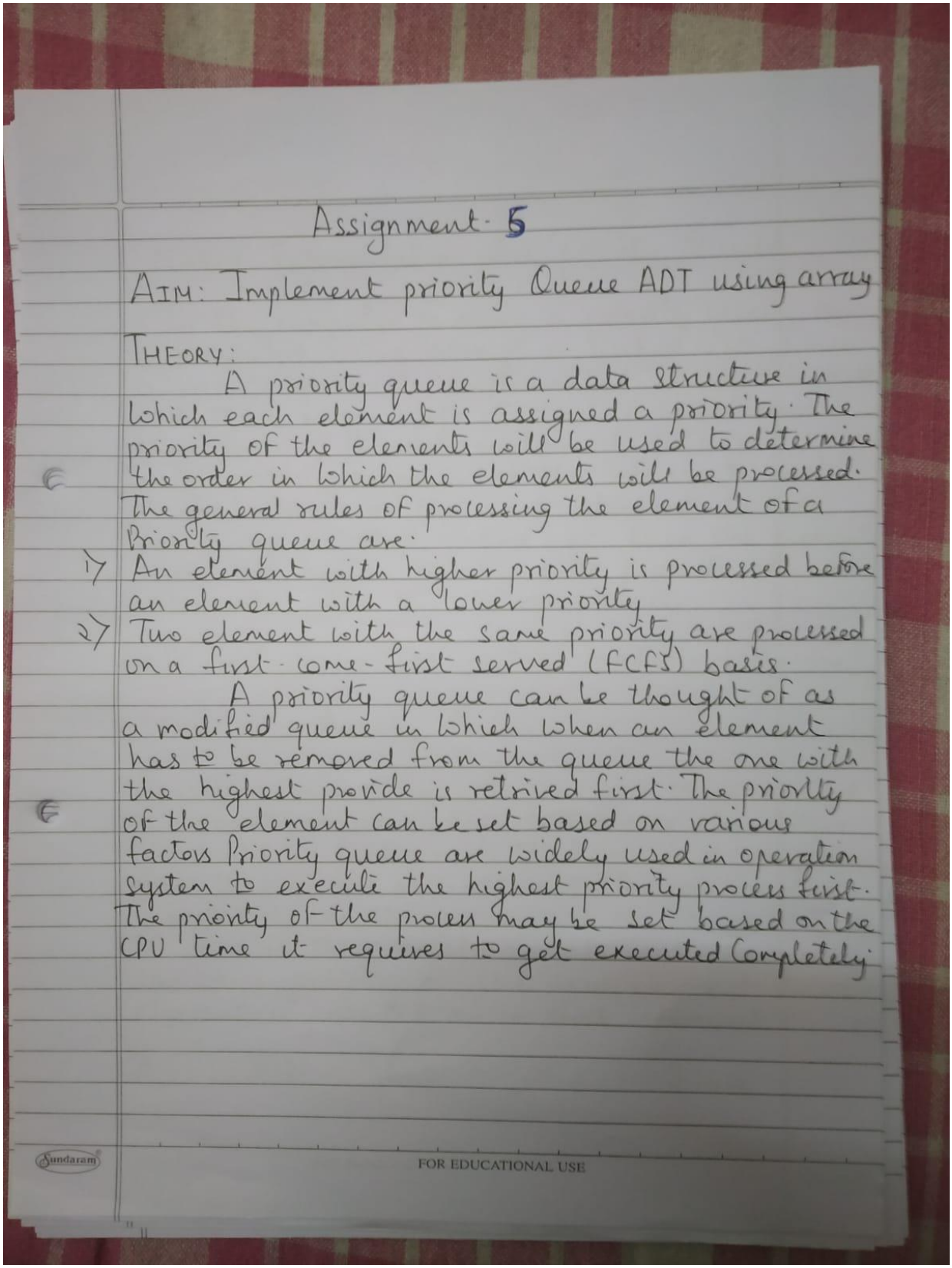
COMPUTER ENGINEERING

DS ODD SEM 2021-22/EXPERIMENT 5

NAME:- GAURAV AMARNANI (D7A, 67)



Assignment- 5

AIM: Implement priority Queue ADT using array

THEORY:

A priority queue is a data structure in which each element is assigned a priority. The priority of the elements will be used to determine the order in which the elements will be processed. The general rules of processing the element of a Priority queue are:

1) An element with higher priority is processed before an element with a lower priority
2) Two element with the same priority are processed on a first-come-first served (FCFS) basis.

A priority queue can be thought of as a modified queue in which when an element has to be removed from the queue the one with the highest provide is retrived first. The priority of the element can be set based on various factors Priority queue are widely used in operation system to execute the highest priority process first. The priority of the process may be set based on the CPU time it requires to get executed Completely

Sundaram

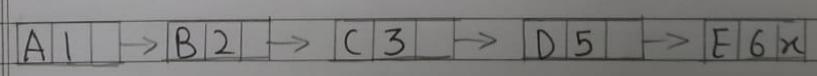
FOR EDUCATIONAL USE

Linked Representation of a priority Queue

In the computer memory a priority queue can be presented using arrays or linked lists. When a priority queue is implemented using a linked list, then every node of the list will have three parts a) The information of data part b) The priority number of the element and c) The address of the next element. If we are using a sorted linked list, then the element with the higher priority will precede the element with the lower priority

Priority Queue

Insertion when a new element has to be inserted in a priority queue. we have to traverse the entire list until we find a node that has a priority lower than that of the new element. The new node is inserted before the node with the lower priority. However if there exists, an element that has the same priority as the new element. The new element is inserted after that element



Deletion: Deletion is a very simple process in this case. The fixed node of the list will be deleted and the data of that node will be processed first

Conclusion:

In this experiment we implemented priority queue adt using array program and learnt various applications of the following

PROGRAM:

```
#include<stdio.h>
#define N 50
int Q[N],Pr[N];
int r = -1,f = -1;
void enqueue(int data,int p) {
int i;
if((f==0)&&(r==N-1))
else {
printf("Queue is full");
if(f==-1) {
f = r = 0;
Q[r] = data;
Pr[r] = p;
}
else if(r == N-1) {
for(i=f;i<=r;i++) {
Q[i-f] = Q[i];
Pr[i-f] = Pr[i];
r = r-f;
f = 0;
else
Q[i+1] = Q[i];
Pr[i+1] = Pr[i];
break;
Q[i+1] = data;
Pr[i+1] = p;
r++;
}
}
}
else {
Q[i+1] = Q[i];
Pr[i+1] = Pr[i];
}
}
void print() {
int i;
for(i=f;i<=r;i++) {
printf("\nElement = %d\tPriority = %d",Q[i],Pr[i]);
}
}
int dequeue() {
if(f == -1) {
}
else {
printf("Queue is Empty");
printf("deleted Element = %d\t Its Priority = %d",Q[f],Pr[f]);
if(f==r)
f = r = -1;
else
}
f++;
```

```
return(0);
}

void main() {
int opt,n,i,data,p;
printf("Enter Your Choice:-");
do{
printf("\n\n1 for Insert the Data in Queue\n2 for show the Data in Queue \n3 for Delete the
data from the Queue\n0 for Exit");
scanf("%d",&opt);
switch(opt){
case 1:
printf("\nEnter the number of data");
scanf("%d",&n);
printf("\nEnter your data and Priority of data");
i=0;
while(i<n) {
scanf("%d %d",&data,&p);
enqueue(data,p);
i++;
}
break;
case 2:
print();
break;
case 3:

dequeue();
break;
case 0:
break;
default:
printf("\nIncorrect Choice");
}
} while(opt!=0);
}
```

OUTPUT:-

```
C:\TURBOC3\BIN>TC
Enter Your Choice:-

1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
1

Enter the number of data 3

Enter your data and Priority of data
10 17
20 18
30 69


1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
2




1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
2

Element = 30    Priority = 69
Element = 20    Priority = 18
Element = 10    Priority = 17

1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
3
deleted Element = 30     Its Priority = 69

1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit

_
```