```
#include<stdio.h>
#include<stdlib.h>
void main(){
	int i,max;
	int Q[100];
	int front = -1;
	int rear = -1;
	int item;
	int ch=1;
	printf("QUEUE OPERATIONS\n");
	for(i=0;i<17;i++){
		printf("%c",'-');
	}
	printf("\n");
	printf("Enter size of the Queue = ");
	scanf("%d",&max);
	while(ch<5){
		printf("1.Enqueue\n2.Dequeue\n3.Print the Queue\n4.exit\n\n");
		printf("Enter your choice = ");
		scanf("%d",&ch);
		switch(ch){
			case 1 : if(rear == max-1){
					printf("Queue overflow/Queue is full\nredirecting to
main menu...\n\n");
					break;
				}
				else if(front==-1 && rear==-1){
					front=rear=0;
					printf("Enter the number to insert first = ");
					scanf("%d",&item);
					Q[rear]=item;
				}
				else{
		rear = rear+1;
		printf("Enter the number to insert = ");
		scanf("%d",&item);
		Q[rear] = item;
				}
				break;
			case 2 : if(front==-1 && rear==-1){
					printf("Queue underflow/Queue is empty\nredirecting
to main menu...\n\n");
				}
		else if(front==rear){
		    item = Q[front];
		    printf("The last element deleted = %d\n",Q[front]);
		    front=rear=-1;
		  }
				else{
		    item = Q[front];
		    printf("The element that deleted  = %d\n",Q[front]);
		    front = front+1;
```

```
                                        }
                                        break;
                        case 3 : if(front == -1){
                                                printf("Queue is empty\nredirecting to main menu...\n\
n");
                                                break;
                                        }
                                        else{
                                                printf("The current queue\n");
                                                for(i=front;i<=rear;i++){
                                                        printf("%d\n",Q[i]);
                                                }
                                        }
                                        printf("\n");
                                        break;
                        case 4 : printf("exiting the program...\n");
                                        exit(0);
                        default: printf("Something went wrong !!!\nprogram terminated...\n");
                                        exit(0);

                }
        }
}


#include<stdio.h>
#include<stdlib.h>
int main(){
        int i,max;
        int Q[100];
        int front = -1;
        int rear = -1;
        int item;
        int ch=1;
        printf("QUEUE OPERATIONS\n");
        for(i=0;i<16;i++){
                printf("%c",'-');
        }
        printf("\n");
        printf("Enter size of the Queue = ");
        scanf("%d",&max);
        while(ch<5){
                printf("1.Enqueue\n2.Dequeue\n3.Print the Queue\n4.exit\n\n");
                printf("Enter your choice = ");
                scanf("%d",&ch);
                switch(ch){
                        case 1 : if(rear == max-1){
                                                printf("Queue overflow/Queue is full\nredirecting to
main menu...\n\n");
                                                break;
                                        }
                                        else if(front==-1 && rear==-1){
```

```
                                        front=rear=0;
                                        printf("Enter the number to insert first = ");
                                        scanf("%d",&item);
                                        Q[rear]=item;
                                }
                                else{
                rear = rear+1;
                printf("Enter the number to insert = ");
                scanf("%d",&item);
                Q[rear] = item;
                                }
                                break;
                        case 2 : if(front==-1 && rear==-1){
                                        printf("Queue underflow/Queue is empty\nredirecting
to main menu...\n\n");
                                }
                else if(front==rear){
                                        item = Q[front];
                                        printf("The last element deleted = %d\n",Q[front]);
                                        front=rear=-1;
                }
                                else{
                                        item = Q[front];
                                        printf("The element that deleted  = %d\n",Q[front]);
                                        front = front+1;
                                }
                                break;
                        case 3 : if(front == -1){
                                        printf("Queue is empty\nredirecting to main menu...\n\
n");
                                        break;
                                }
                                else{
                                        printf("The current queue\n");
                                        for(i=front;i<=rear;i++){
                                                printf("%d\n",Q[i]);
                                        }
                                }
                                printf("\n");
                                break;
                        case 4 : printf("exiting the program...\n");
                                exit(0);
                        default: printf("Something went wrong !!!\nprogram terminated...\n");
                                exit(0);

                }
        }
        return 0;
}
```

```
#include<stdio.h>
#include<stdlib.h>
int main(){
	int i,max;
	int op=1;
	int front=-1;
	int rear=-1;
	int item;
	int Q[max];
	printf("CIRCULAR QUEUE OPERATIONS\n");
	for(i=0;i<25;i++){
		printf("%c",'-');
	}
	printf("\n");
	printf("Enter size of circular Queue = ");
	scanf("%d",&max);
	while(op<4){
		printf("1.Enqueue\n2.Dequeue\n3.Print CircularQ\n4.exit\n\n");
		printf("Choice = ");
		scanf("%d",&op);
		switch(op){
			case 1 : if((rear+1)%max == front){
					printf("Queue overflow/full\n");
					break;
				}
				else if(front==-1){
					front=rear=0;
					printf("Enter the number to insert first = ");
					scanf("%d",&item);
					Q[rear]=item;
				}
				else{
					rear = (rear+1)%max;
					printf("Enter the element to insert = ");
					scanf("%d",&item);
					Q[rear]=item;
				}
				break;
			case 2 : if(front==-1){
					printf("Queue underflow/empty\n");
				}
				else if(front==rear){
					item=Q[front];
					printf("The last element deleted = %d\n",Q[front]);
					front=rear=-1;
				}
				else{
					item=Q[front];
					printf("The element deleted = %d\n",Q[front]);
```

```
                                        front=(front+1)%max;

                                }
                                break;
                        case 3 : if(front==-1){
                                        printf("Queue is empty\nredirecting to main menu...\
n");
                                }
                                else{
                                        /*for(i=front;i<=rear;i++){
                                                printf("%d\n",Q[i]);
                                        }*/
                                        while(front != rear){
                                                printf("%d\n",Q[front]);
                                                front=(front+1)%max;
                                        }
                                        printf("%d\n",Q[rear]);
                                        printf("\n");
                                }
                                break;
                        case 4 : printf("exiting the program...\n");
                                exit(0);
                        default: printf("Something went wrong !!!\nterminating the program...\n");
                                exit(0);
                }
        }
        printf("\n");
        return 0;
}
#include<stdio.h>
#include<stdlib.h>
struct node{
        int data;
        struct node*link;
}*head,*p,*temp;
void main(){
        head = NULL;
        int ch=1;
        printf("choice = ");
        scanf("%d",&ch);
        p = (struct node*)malloc(sizeof(struct node)); //return a void pointer
        printf("Enter the data = ");
        scanf("%d",&p->data);
        p->link=0;
        if(head == NULL){
                head = temp = p;
        }
        else{
                temp->link = p;
                temp = p;
        }
```

```
#include<stdio.h>
#include<stdlib.h>
void main(){
	int i,max;
	int DQ[100];
	int front = -1;
	int rear = -1;
	int item;
	int ch = 1;
	printf("DOUBLE ENDED  QUEUE OPERATIONS\n");
	for(i=0;i<30;i++){
		printf("%c",'-');
	}
	printf("\n");
	printf("Enter size of Dequeue = ");
	scanf("%d",&max);
	while(ch<4){
		printf("1.Enque\n2.Deque\n3.Print the current Queue\n4.press any other key to exit\
n");
		printf("Enter your choice = ");
		scanf("%d",&ch);
		switch(ch){
			case 1 : printf("you have selected Enqueue operations\n");
				if(rear == max-1){
					printf("DEQ overflow/full\n");
					break;
				}else if(front == 0){
					front == max-1;
					printf("Enter the element to insert from last one by one
= ");
					scanf("%d",&item);
					DQ[front] =item;
					front = front-1;
				}else if(front == -1){
					front = 0;
					printf("Enter number to insert at first position = ");
					scanf("%d",&item);
					DQ[front] = item;
				}else{
					printf("Enter the number to insert = ");
					scanf("%d",&item);
					DQ[front] = item;
					rear = rear +1;
				}break;
			case 2 :printf("You have selected Dequeue operations\n");
				if(front == rear == -1){
					printf("DQE underflow/empty\n");
					break;
				}else if(front == rear){
					printf("%d is the last element in this DQE\n",item);
```

```
                                        item = DQ[front];
                                        printf("The DQE is know empty\n");
                                        front = rear = -1;
                                }else if(front == rear == -1){
                                        printf("DQE underflow/empty\n");
                                        printf("Redirecting to main menu\n");
                                        break;
                                }else if(rear == 0){
                                        printf("The element deleted = %d",item);
                                        item = DQ[rear];
                                        rear = max-1;
                                }else{
                                        printf("The element deleted = %d",item);
                                        item = DQ[rear];
                                        rear = rear-1;
                                }break;
                        case 3 :if(front == rear == -1){
                                        printf("DQE underflow/empty\n");
                                        printf("Redirecting to main menu\n");
                                        break;
                                }else{
                                        for(i=front;i<=rear;i++){
                                                printf("a%d = %d",i,DQ[i]);
                                        }
                                        printf("\n");
                                }break;
                        default: printf("exiting the program\n");
                                exit(0);
                }
        }
}



#include<stdio.h>
#include<stdlib.h>
int item,max,top=-1;
void push(int stack[max]){
   if(top ==max-1){
      printf("stack overflow/stack is full\n");
      exit(0);
   }
   else{
      printf("Enter the number to push = ");
      scanf("%d",&item);
      top=top+1;
      stack[top]=item;
   }
}
void pop(int stack[max]){
   if(top == -1){
      printf("stack underflow/stack is empty\n");
```

```
        exit(0);
    }
    else{
        item=stack[top];
        top=top-1;
        printf("The element that poped = %d\n",item);
    }
}
void print(int stack[max]){
    if(top == -1){
        printf("stack underflow/stack is empty\n");
        exit(0);
    }
    else{
        printf("The current stack\n");
        for(int i=0;i<=top;i++){
            printf("%d \n",stack[i]);
        }
        printf("\n");
    }
}
void main(){
            int op=1;
            printf("STACK OPERATIONS\n");
            printf("Enter size of stack = ");
            scanf("%d",&max)
            int stack[max];
            while(op<4){
                        printf("1.push operation\n2.pop operation\n3.print\n4.exit\n");
                        printf("Choice = ");
                        scanf("%d",&op);
                        switch(op){
            case 1 : push(stack);
                    break;
            case 2 : pop(stack);
                    break;
            case 3 : print(stack);
                    break;
            default: printf("Something wrong!!!\nexiting the program\n");
                                    exit(0);
        }
    }
}

#include<stdlib.h>
#include <stdio.h>
#define max 100
typedef struct
{
            int row ;
            int col ;
            int value ;
```

```
}sparce;
sparce A[max],B[max],C[max];
void main()
{
	int i,j,r1,r2,c1,c2,x,n=1,m=1,sum=0,k=1,c=0;;
	printf("No of rows of 1st = ");
	scanf("%d",&r1);
	printf("No of cols of 1st = ");
	scanf("%d",&c1);
	A[0].row=r1;
	A[0].col=c1;
	printf("No of rows of 2nd = ");
	scanf("%d",&r2);
	printf("No of rows of 2nd = ");
	scanf("%d",&c2);
	B[0].row=r2;
	B[0].col=c2;
	if(r1!=r2 || c1!=c2){
		printf("Matrix addition not poosible !!!\nexiting the program\n!");
		exit(0);
	}

	printf("Enter the elements of first matrix\n");
	for (i=0;i<r1;i++){
		for (j=0;j<c1;j++){
			scanf("%d",&x);
			if(x!=0){
				A[m].row=i;
				A[m].col=j;
				A[m].value=x;
				m++;
			}
		}
	}
	A[0].value=m-1;
	printf("Tuple form of 1st matrix\n");
	for(i=0;i<m;i++){
		printf("%d  %d  %d \n",A[i].row,A[i].col,A[i].value);
	}
	printf("\nEnter the elements of Second matrix\n");
	for (i=0;i<r2;i++){
		for (j=0;j<c2;j++){
			scanf("%d",&x);
			if(x!=0){
				B[n].row=i;
				B[n].col=j;
				B[n].value=x;
				n++;
			}
		}
	}
	B[0].value=n-1;
```

```
printf("Tuple form of 2nd matrix\n");
for(j=0;j<n;j++){
	printf("%d  %d  %d \n",B[j].row,B[j].col,B[j].value);
}
i=1;
j=1;
int p=A[0].value;
int q=B[0].value;
while(i<=p && j<=q){
	if(A[i].row<B[j].row || A[i].col<B[j].col){
		C[k].col=A[i].col;
		C[k].row=A[i].row;
		C[k].value=A[i].value;
		i++;
		k++;
		c++;
	}
	else if(A[i].row>B[j].row || A[i].col>B[j].col){
		C[k].col=B[j].col;
		C[k].row=B[j].row;
		C[k].value=B[j].value;
		j++;
		k++;
		c++;
	}
	else{
		sum=A[i].value + B[j].value;
		if(sum!=0){
			C[k].col=A[i].col;
			C[k].row=A[i].row;
			C[k].value=sum;
			i++;
			k++;
			c++;
			j++;
		}
		else{
			i++;
			j++;
		}
	}
}
while(i<=p){
	C[k].col=A[i].col;
	C[k].row=A[i].row;
	C[k].value=A[i].value;
	i++;
	k++;
	c++;
}
while(j<=q){
	C[k].col=B[j].col;
```

```
                C[k].row=B[j].row;
                C[k].value=B[j].value;
                j++;
                k++;
                c++;
        }
        C[0].value=c;
        C[0].row=r1;
        C[0].col=c1;
        printf("Added matrix is\n");
        for(i=0;i<k;i++){
                printf("%d  %d  %d \n",C[i].row,C[i].col,C[i].value);
        }
}


#include<stdio.h>
#define max 100
typedef struct{
        int row;
        int col;
        int value;
}sparse;
sparse A[max];
sparse B[max];
void main(){
        int a[100][100],i,j,r,c,k=1,p=1;
        printf("SPARSE MATRIX\n");
        printf("Enter no of rows = ");
        scanf("%d",&r);
        printf("Enter no of columns = ");
        scanf("%d",&c);
        printf("Enter the array elements\n");
        for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                        printf("a%d%d = ",i,j);
                        scanf("%d",&a[i][j]);
                }
        }
        printf("The entered matrix is\n");
        for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                        printf("%d  ",a[i][j]);
                }
                printf("\n");
        }
        A[0].row = r;
        A[0].col = c;
        for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                        if(a[i][j] != 0){
                                A[k].row = i;
```

```
                        A[k].col = j;
                        A[k].value = a[i][j];
                        k++;
                    }
                }
        }
        A[0].value = k-1;
        printf("The sparse matrix is\n");
        for(i=0;i<k;i++){
                printf("%d  %d  %d  \n",A[i].row, A[i].col, A[i].value);
        }
        printf("The transpose form\n");
        for(i=0;i<k;i++){
                printf("%d  %d  %d  \n",A[i].col, A[i].row, A[i].value);
        }
        B[0].row = A[0].col;
        B[0].col = A[0].row;
        B[0].value = A[0].value;
        for(i=0;i<=A[0].col;i++){
                for(j=1;j<=A[0].value;j++){
                        if( A[j].col == i){
                                B[p].col = A[j].row;
                                B[p].row = A[j].col;
                                B[p].value = A[j].value;
                                p++;
                        }
                }
        }
        printf("The ordered transpose form\n");
        for(i=0;i<=B[0].value;i++){
                printf("%d  %d  %d  \n",B[i].row,B[i].col,B[i].value);
        }
}



#include<stdio.h>
#define max 100
typedef struct pol{
        int coef;
        int exp;
}pol;
pol A[max];
void main(){
        int sA=0,sB,sC,fA,fB;
        int i,c,p,q;
        printf("POLYNOMIAL ADDITION\n");

        printf("Enter the no of terms in pol 1 = ");
        scanf("%d",&p);
        printf("Enter the no of terms in pol 2 = ");
        scanf("%d",&q);
```

```
sA=0;
fA=p-1;
sB=p;
fB=p+q-1;
sC=p+q;
for(i=0;i<p;i++){
        printf("Coef of pol 1 at a%d = ",i);
        scanf("%d",&A[i].coef);
        printf("Exp of pol 1 at  a%d = ",i);
        scanf("%d",&A[i].exp);
}
for(i=p;i<p+q;i++){
        printf("Coef of pol 2 at a%d = ",i);
        scanf("%d",&A[i].coef);
        printf("Exp of pol 2 at  a%d = ",i);
        scanf("%d",&A[i].exp);
}

while(sA<=fA && sB<=fB){
        if(A[sA].exp > A[sB].exp){
                A[sC].exp = A[sA].exp;
                A[sC].coef = A[sA].coef;
                sA++;
                sC++;
        }
        else if(A[sA].exp < A[sB].exp){
                A[sC].exp = A[sB].exp;
                A[sC].coef = A[sB].coef;
                sB++;
                sC++;
        }
        else{
                c = A[sA].coef + A[sB].coef;
                if(c != 0){
                        A[sC].exp = A[sB].exp;
                        A[sC].coef = c;
                        sC++;
                }
                sA++;
                sB++;
        }
}
while(sA<=fA){
        A[sC].exp = A[sA].exp;
        A[sC].coef = A[sA].coef;
        sA++;
        sC++;
}
while(sB<=fB){
        A[sC].exp = A[sB].exp;
        A[sC].coef = A[sB].coef;
        sB++;
```

```
                sC++;
        }

        printf("The first polynomial  = ");
        for(i=0;i<p;i++){
                printf("%d x ^%d + ",A[i].coef, A[i].exp);
        }
        printf("\n");
        printf("The second polynomial = ");
        for(i=p;i<p+q;i++){
                printf("%d x ^%d + ",A[i].coef, A[i].exp);
        }
        printf("\n");
        printf("The values of index positions\n sA=%d\n fA=%d\n sB=%d\n fB=%d\n sC=%d\
n",sA,fB,sB,fB,sC);
        printf("The added polynomial = ");
        for(i=p+q;i<sC;i++){
                printf("%d x^%d + ",A[i].coef, A[i].exp);
        }
        printf("\n");
}




#include<stdio.h>
#include<stdlib.h>
void main(){
        int i,j,p,x,temp,n,op1=1,op2=1,op3=1,a[100],sum=0,key,flag=0;
        printf("ARRAY OPERATIONS\n");
        for(i=0;i<16;i++){
                printf("%c",'-');
        }
        printf("\n");
        printf("Enter size of array = ");
        scanf("%d",&n);
        printf("Enter the array elements\n");
        for(i=0;i<n;i++){
                printf("a%d = ",i);
                scanf("%d",&a[i]);
        }
        printf("The entered array\n");
        for(i=0;i<n;i++){
                printf("%d\n",a[i]);
        }
        printf("\n");
        while(op1<7){
                printf("1.Insertion at specific position\n2.Deletion from specific position\
n3.Searching for an element\n4.Sum of array elements\n5.Sorting\n6.exit\n\n");
                printf("Enter your choice = ");
                scanf("%d",&op1);
                switch(op1){
```

```
                case 1 : printf("Enter the element you want to insert = ");
                         scanf("%d",&x);
                         printf("Enter the position(i+1) you want = ");
                         scanf("%d",&p);
                         for(i=n-1;i>=p-1;i--){
                                 a[i+1]=a[i];
                         }
                         a[p-1]=x;
                         printf("The array after insertion of the element\n");
                         for(i=0;i<=n;i++){
                                 printf("a%d = %d\n",i,a[i]);
                         }
                         printf("\n");
                         break;
                case 2 : printf("Enter the position(i+1) of the element = ");
                         scanf("%d",&p);
                         if(p>=n+1 || p<=0){
                                 printf("Deletion is not possible\n");
                         }else{
                                 for(i=p-1;i<n-1;i++){
                                         a[i]=a[i+1];
                                 }
                                 n--;
                                 printf("The array after deletion of the element\n");
                                 for(i=0;i<n-1;i++){
                                         printf("a%d = %d\n",i,a[i]);
                                 }
                         }
                         printf("\n");
                         break;
                case 3 : printf("1.Linear search\n2.Binary search(this works only in a sorted
array)\n");
                         printf("Enter searching choice = ");
                         scanf("%d",&op2);
                         switch(op2){
                                 case 1 : printf("You have selected Linear search\n");
                                          printf("Enter the key = ");
                                          scanf("%d",&key);
                                          for(i=0;i<n;i++){
                                                  if(a[i]==key){
                                                          flag=1;
                                                          break;
                                                  }
                                          }
                                          if(flag==1){
                                                  printf("%d is found at index
position a%d\n",key,i);
                                          }else{
                                                  printf("Element is not found in
the array\n");
                                          }
                                          printf("\n");
```

```
                                                        break;
                                                case 2 : printf("You have selected binary search(in
sorted array otherwise you will get wrong output)\n");
                                                        printf("Enter the key = ");
                                                        scanf("%d",&key);
                                                        int low=0,high=n-1,mid=(low+high)/2;
                                                        while(low <= high){
                                                                if(a[mid]<key){
                                                                        low=mid+1;
                                                                }else if(a[mid]==key){
                                                                        printf("%d is found at the
index position %d\n",key,mid);
                                                                        break;
                                                                }else{
                                                                        high=mid-1;
                                                                        mid=(low+high)/2;
                                                                }
                                                        }
                                                        if(low > high){
                                                                printf("%d is not present in the
array\n",key);
                                                        }
                                                        printf("\n");
                                                        break;
                                          default: printf("invalid choice...back to main menu\n");
                                                        break;
                                        }break;
                        case 4 : for(i=0;i<n;i++){
                                                sum=sum+a[i];
                                         }
                                        printf("The sum of array elements = %d\n",sum);
                                        printf("\n");
                                        break;
                        case 5 : printf("1.Selection sort\n2.Insertion sort\n3.Bubble sort\n");
                                        printf("Sorting choice = ");
                                        scanf("%d",&op3);
                                        switch(op3){
                                                case 1 : printf("You have selected Selection sort\n");
                                                        for(i=0;i<n-1;i++){
                                                                for(j=i+1;j<n;j++){
                                                                        if(a[i]>a[j]){
                                                                                temp=a[i];
                                                                                a[i]=a[j];
                                                                                a[j]=temp;
                                                                        }
                                                                }
                                                        }
                                                        printf("The sorted array using Selection
sort\n");
                                                        for(i=0;i<n;i++){
                                                                printf("a%d = %d\n",i,a[i]);
                                                        }
```

```
                                        printf("\n");
                                        break;
                                case 2 : printf("You have selected Insertion sort\n");
                                         for(i=1;i<n;i++){
                                                temp = a[i];
                                                j=i-1;
                                                while(j>=0){
                                                        if(a[j] > temp){
                                                                a[j+1] = a[j];
                                                                j=j-1;
                                                                a[j+1] = temp;
                                                        }
                                                        else{
                                                                break;
                                                        }
                                                }
                                        }
                                        printf("Sorted array using Insertion sort\
n");
                                        for(i=0;i<n;i++){
                                                printf("a%d = %d\n",i,a[i]);
                                        }
                                        printf("\n");
                                        break;
                                case 3 : printf("You have selected Bubble sort\n");
                                         for(i=0;i<n;i++){
                                                for(j=0;j<n-i-1;j++){
                                                        if(a[j]>a[j+1]){
                                                                temp=a[j];
                                                                a[j]=a[j+1];
                                                                a[j+1]=temp;
                                                        }
                                                }
                                        }
                                        printf("The sorted array using Bubble
sort\n");
                                        for(i=0;i<n;i++){
                                                printf("a%d = %d\n",i,a[i]);
                                        }
                                        printf("\n");
                                        break;
                                default:printf("invalid choice...back to main menu\n\
n");
                                        break;
                                }break;
                        case 6 : printf("exiting the program...\n");
                                 exit(0);
                        default: printf("invalid key !!! Program terminated...\n");
                                 exit(0);
                }
        }
}
```