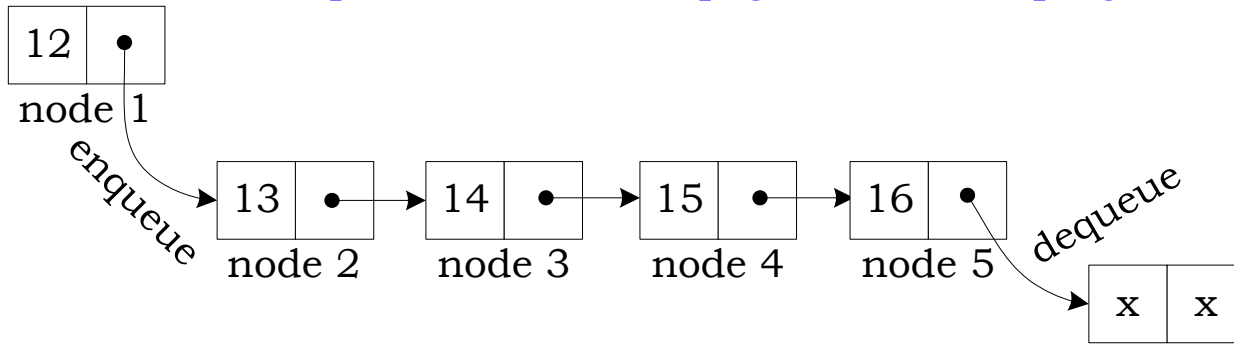


C PROGRAM CODE FOR QUEUE DATA STRUCTURE

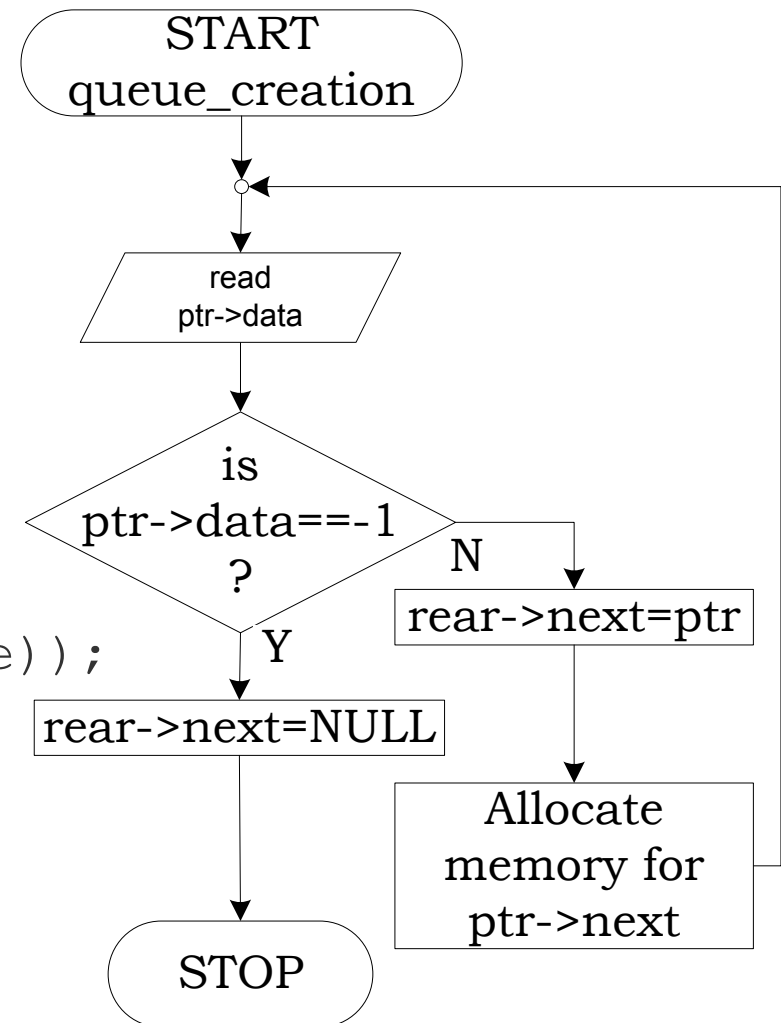
<http://radhikasree.hubpages.com/hub/Cprogramcodeforqueuestructure>



```
void createqueue(que *ptr)
{
    printf("Enter the data:");
    scanf("%d", &ptr->data);

    if( (ptr->data) != -1 )
    {
        rear->next=ptr;
        rear=ptr;

        ptr->next=(que *)malloc(sizeof(que));
        createqueue(ptr->next);
    }
    else
        rear->next=NULL;
}
```

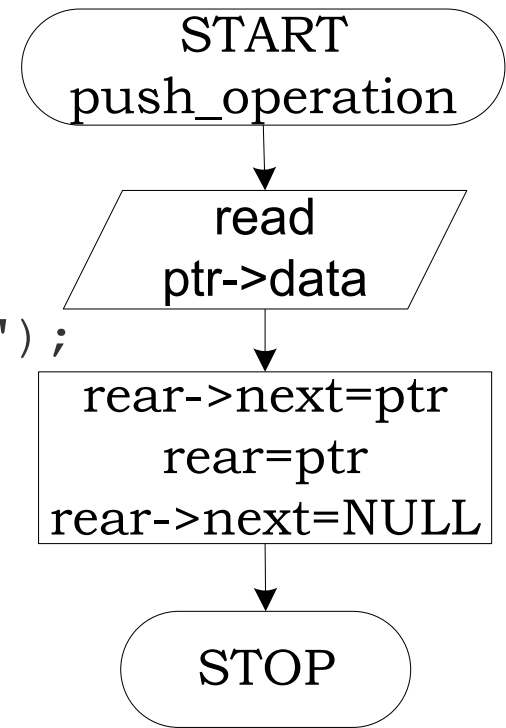


```

void push()
{
    que *ptr;
    ptr=(que *)malloc(sizeof(que));

    printf("\nEnter data to push into the queue:");
    scanf("%d",&ptr->data);
    rear->next=ptr;
    rear=ptr;
    rear->next=NULL;
}

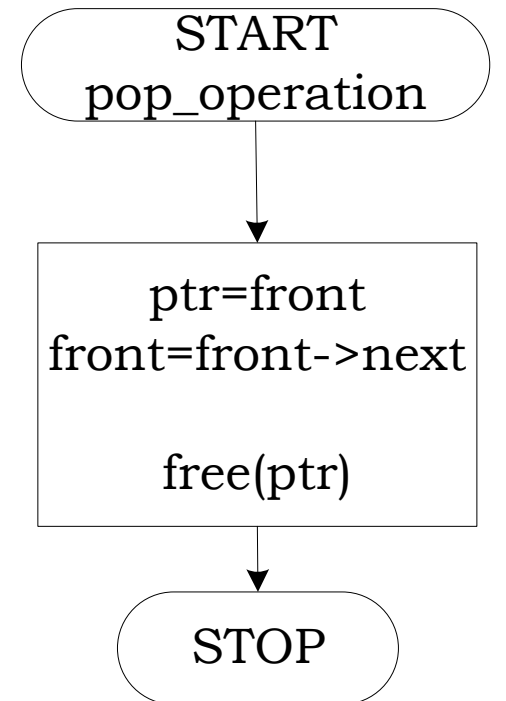
```



```

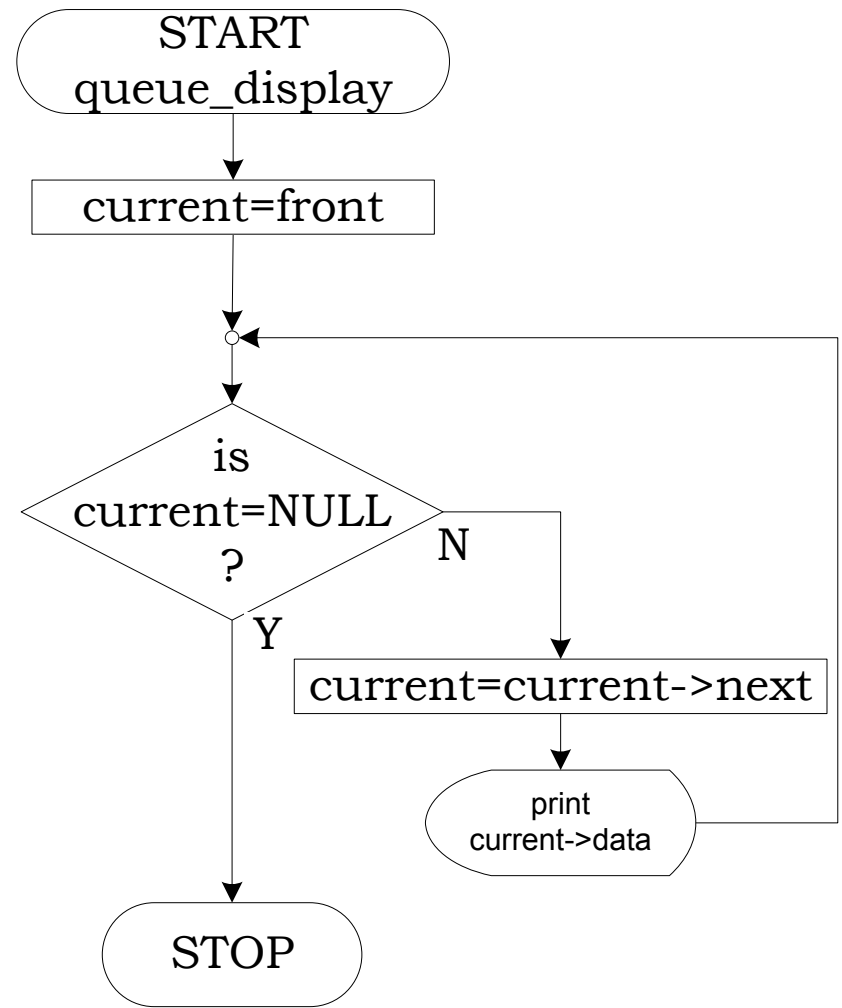
void pop()
{
    que *ptr;
    ptr=front;
    front=front->next;
    free(ptr);
}

```



```
void display()
{
    que *cur;
    printf("\n\t");
    cur=front;

    while (cur)
    {
        printf("%d---->", cur->data);
        cur=cur->next;
    }
}
```



```
// C program code to create a queue, push an element into the queue  
// (enqueue) and pop an element out of the queue (dequeue) and display  
// the values entered in the queue  
//  
#include<stdio.h>  
#include<conio.h>  
#include<malloc.h>  
  
struct queue  
{  
    int data;  
    struct queue *next;  
};  
  
typedef struct queue que;  
que *front, *rear;  
  
void main()  
{  
    int ch;  
  
    void createqueue(que *);  
    void push();  
    void pop();  
    void display();  
    que *ptr;  
  
    printf("Create a queue(-1 to end):\n\n");
```

```
ptr=(que *)malloc(sizeof(que));

front=ptr;
rear=ptr;

createque(ptr);

printf("Given queue is\n");
printf("-----\n");

display();
display();

do
{
    printf("\nQueue Operations");
    printf("\n-----");
    printf("\n1. Push");
    printf("\n2. Pop");
    printf("\n3. Exit");

    printf("\n\n\tEnter your choice:");
    scanf("%d", &ch);

switch(ch)
{
case 1:push();
    printf("\n\tQueue after push operation is:\n\n");
```

```
display();  
break;
```

```
case 2: pop();  
printf("\n\tQueue after pop operation is:\n\n");  
display();  
break;
```

```
case 3: exit(1);  
}
```

```
display();  
}
```

```
while(ch!=3);  
getch();  
}
```

```
void createque(que *ptr)  
{
```

```
printf("Enter the data:");  
scanf("%d", &ptr->data);
```

```
if((ptr->data) != -1)  
{
```

```
rear->next=ptr;  
rear=ptr;
```

```
ptr->next=(que *)malloc(sizeof(que));
```

```
        createque(ptr->next);
    }
    else
        rear->next=NULL;
}
```

```
void push()
{
    que *ptr;
    ptr=(que *)malloc(sizeof(que));

    printf("\n\tEnter the data to push into the queue:");
    scanf("%d",&ptr->data);
    rear->next=ptr;
    rear=ptr;
    rear->next=NULL;
}
```

```
void pop()
{
    que *ptr;
    ptr=front;
    front=front->next;
    free(ptr);
}
```

```
void display()
```

```
{
    que *cur;
    printf("\n\t");
    cur=front;

    while (cur)
    {
        printf("%d---->", cur->data);
        cur=cur->next;
    }
    getch();
}
```