1.void test(int sum)

{

int x;

scanf("%d",&x);

if (x==0)

sum=0;

else

{

do

{

sum+=x;

scanf("%d",&x);

}while (x!=0);

}

printf("%d ",sum);

}

2.

队列初始化

struct node \* initQueue(struct node \* rear) //队列初始化

{

rear=(struct node \*)malloc(sizeof(Node)); //定义尾指针

if(!rear)

return 0;

else

{

rear->next=rear; //使队列循环

}

return rear;

}

入队列

struct node \* enQueue(struct node \* rear,ElemType e) //入队列

{

struct node \* p=(struct node \*)malloc(sizeof(Node));

p->data=e;

p->next=rear->next;

rear->next=p;

rear=p; //现在尾指针指向p

count++;

return rear;

}

出队列

struct node \* deQueue(struct node \* rear) //出队列

{

int e;

struct node \* p;

if(rear==rear->next)

printf("队列为空");

else{

p=rear->next->next;

e=p->data;

rear->next->next=p->next;

if(rear==p)

{

rear=rear->next;

}

free(p);

}

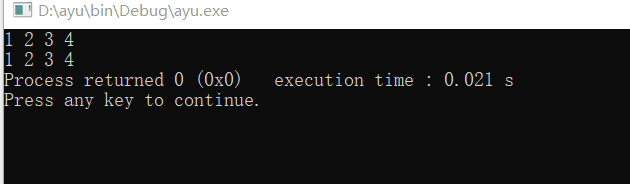
printf("%d",e);

count--;

return rear;

}

程序运行截图



源代码

#include <stdio.h>

#include <malloc.h>

int count=0;

typedef int ElemType;

typedef struct node{

ElemType data;

struct node \*next;

}Node;

struct node \* initQueue(struct node \* rear) //队列初始化

{

rear=(struct node \*)malloc(sizeof(Node)); //定义尾指针

if(!rear)

return 0;

else

{

rear->next=rear; //使队列循环

}

return rear;

}

struct node \* enQueue(struct node \* rear,ElemType e) //入队列

{

struct node \* p=(struct node \*)malloc(sizeof(Node));

p->data=e;

p->next=rear->next;

rear->next=p;

rear=p; //现在尾指针指向p

count++;

return rear;

}

struct node \* deQueue(struct node \* rear) //出队列

{

int e;

struct node \* p;

if(rear==rear->next)

printf("队列为空");

else{

p=rear->next->next;

e=p->data;

rear->next->next=p->next;

if(rear==p)

{

rear=rear->next;

}

free(p);

}

printf("%d",e);

count--;

return rear;

}

int main()

{

struct node \* rear=initQueue(rear);

struct node \*p;

struct node \*q;

int i;

for(i=0;i<4;i++)

{

rear=enQueue(rear,i+1);

}

p=rear->next->next;

for(i=0;i<4;i++)

{

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

p=p->next;

}

printf("\n");

q=rear->next->next;

for(i=0;i<4;i++)

{

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

q=q->next;

}

return 0;

}