**Bài 4: Cấu trúc ngăn xếp (Stack)**

**Stack cài đặt bằng mảng**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#define N 20

typedef struct node

{ long maso;

char hoten[40];

char diachi[100];

}node;

typedef struct stack

{

int top;

node list[N];

}stack;

void khoitao(stack &s)

{

s.top=-1;

}

int empty(stack s)

{

return s.top==-1?1:0;

}

int full(stack s)

{

return s.top==N-1?1:0;

}

void push(stack &s,node x)

{

if(!full(s))

s.list[++s.top]=x;

}

node pop(stack &s)

{ node x;

if(!empty(s))

x=s.list[s.top--];

return x;

}

int main()

{

int n;stack s;node x;

int t1; char \*t2 ; char \*t3 ;

printf("\n nhap vao so phan tu:");

scanf("%d",&n);

khoitao(s);

for(int i=0;i<n;i++)

{

printf("\n nhap ma so:");

scanf("%d",&x.maso);

printf("\n nhap ho ten:");

fflush(stdin);

gets(x.hoten);

printf("\n nhap dia chi :");

fflush(stdin);

gets(x.diachi);

push(s,x);

}

while(!empty(s))

{

x=pop(s);

printf("\nMa so:%d Ho ten:%s Dia chi:%s",x.maso,x.hoten,x.diachi);

/\* printf("\n ma so:%d",x.maso);

printf("\n ho ten:%s",x.hoten);

printf("\n dia chi:%s",x.diachi);

\*/

}

}

**Stack cài đặt bằng danh sách liên kết**

#include <stdio.h>

#include<conio.h>

typedef struct sinhvien

{

char hoten[40];

int tuoi;

}sinhvien;

typedef struct node

{

sinhvien data;

node \*link;

}node;

typedef struct stack

{

node \*top;

}stack;

void khoitao(stack &s)

{

s.top=NULL;

}

int empty(stack s)

{

return s.top==NULL?1:0;

}

void push(stack &s,sinhvien x)

{

node \*p;

p=new(node);

p->data=x;

p->link=s.top;

s.top=p;

}

sinhvien pop(stack &s)

{

node \*p;

if(!empty(s))

p=s.top;

s.top=p->link;

return p->data;

}

int main()

{

int n;stack s;sinhvien x;

printf("\nmhap vao so phan tu:");

scanf("%d",&n);

khoitao(s);

for(int i=0;i<n;i++)

{

printf("\nnhap ho ten:");

fflush(stdin);gets(x.hoten);

printf("\nnhap tuoi:");

scanf("%d",&x.tuoi);

push(s,x);

}

while(!empty(s))

{

x=pop(s);

printf("\nho ten :%s",x.hoten);

printf("\ntuoi:%d",x.tuoi);

}

getch();

}

**Bài 5: Cấu trúc hàng đợi (Queue)**

**Queue cài đặt bằng mảng**

#include"stdio.h"

#include"conio.h"

#define N 20

typedef struct node

{

char hoten[40];

int tuoi;

}node;

typedef struct Queu

{

int front,rear;

node list[N];

}Queu;

void khoitao(Queu &q)

{

q.front=q.rear=-1;

}

int empty(Queu q)

{

if((q.front==-1)&&(q.rear==-1))

return 1;

else

return 0;

}

int full(Queu q)

{

if(q.front==0&&q.rear==N-1)

return 1;

if(q.front==q.rear+1)

return 1;

else

return 0;

}

void enqueu(Queu &q,node x)

{

if(!full(q))

{

if(empty(q))

q.front=q.rear=0;

else

if(q.rear==N-1)

q.rear=0;

else

q.rear=q.rear+1;

q.list[q.rear]=x;

}

}

node dequeu(Queu &q)

{

if(!empty(q))

{

node t=q.list[q.front];

if(q.front==q.rear)

khoitao(q);

else

if(q.front==N-1)

q.front=0;

else

q.front=q.front+1;

return t;

}

}

int main()

{

int n; node x;Queu q;

printf("nhap so phan tu:");

scanf("%d",&n);

khoitao(q);

for(int i=1;i<=n;i++)

{

printf("nhap ho ten:");

fflush(stdin);

gets(x.hoten);

printf("nhap tuoi:");

scanf("%d",&x.tuoi);

enqueu(q,x);

}

while(!empty(q))

{

x=dequeu(q);

printf("\n hotenL:%s",x.hoten);

printf("\n tuoi:%d",x.tuoi);

}

getch();

}

**Queue cài đặt bằng danh sách liên kết**

#include<stdio.h>

#include<conio.h>

# define N 20

typedef struct sinhvien

{

char hoten[40];

int tuoi;

}sinhvien;

typedef struct node

{

sinhvien data;

node \*link;

}node;

typedef struct queu

{

node \*front;

node \*rear;

}queu;

void khoitao(queu &q)

{

q.front=q.rear=NULL;

}

int empty(queu q)

{

if(q.front==NULL)

return 1;

else

return 0;

}

void enqueu(queu &q, node\*new\_ele)

{

if(q.front==NULL)

{

q.front=new\_ele;

q.rear=q.front;

}

else

{

q.rear->link=new\_ele;

q.rear=new\_ele;

}

}

sinhvien dequeu(queu &q)

{

node \*p;sinhvien x;

if(!empty(q))

{

p=q.front;

x=p->data;

q.front=p->link;

delete(p);

}

return x;

}

int main()

{

int n;

sinhvien x;

queu q;

node \*new\_ele;

printf("\n nhap so phan tu:");

scanf("%d",&n);

khoitao(q);

for(int i=1; i<=n; i++)

{

printf("\n nhap hoten:");

fflush(stdin);

gets(x.hoten);

printf("\n nhap tuoi:");

scanf("%d", &x.tuoi);

new\_ele=new(node);

new\_ele->data=x;

new\_ele->link=NULL;

enqueu(q,new\_ele);

}

while(!empty(q))

{

x=dequeu(q);

printf("\n hoten:%s", x.hoten);

printf("\n tuoi:%d", x.tuoi);

}

getch();

}