BÀI THỰC HÀNH SỐ 5

Nội dung:

- Hàm đệ qui

- Khử đệ qui

**Bài 4.1: Viết các hàm đệ qui đầu và đệ qui đuôi thực hiện các yêu cầu sau:**

**a) Tính S = 1 + 2 + 3 + .. + n (100 < n < 10000)**

#include<stdio.h>

int sumdau(int x);

int sumduoi(int x,int s,int i);

int main()

{

int n,s=0,i=0;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=100)||(n>=10000));

printf("tinh theo de quy dau: %d",sumdau(n));

printf("\ntinh theo de quy duoi: %d",sumduoi(n,s,i));

}

int sumdau(int x)

{

if(x>=1) return x+sumdau(x-1);

}

int sumduoi(int x,int s=0,int i=0)

{

if(i>x) return s;

else return sumduoi(x,s+i,i+1);

}

**b) Tính tổng n phần tử của mảng a (a[0], a[1] ,..., a[n-1]) ( 5 < n < 100)**

#include<stdio.h>

#define max 100

int a[max];

int summangdau(int a[],int x);

int summangduoi(int a[],int x,int sum,int cs);

int main()

{

int n,i,cs=0,sum=0;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=100));

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

{

printf("nhap a[%d]: ",i+1);

scanf("%d",&a[i]);

}

printf("tinh theo de quy dau: %d",summangdau(a,n-1));

printf("\ntinh theo de quy duoi: %d",summangduoi(a,n-1,sum,cs));

}

int summangdau(int a[],int x)

{

if (x==0) return a[x];

else return a[x]+summangdau(a,x-1);

}

int summangduoi(int a[],int x,int sum=0,int cs=0)

{

if (cs>x) return sum;

else return summangduoi(a,x,sum+a[cs],cs+1);

}

**c) Một khách hàng gửi ngân hàng 100.000 với lãi suất 6.5%/năm. Hỏi sau n năm**

**khách hàng thu được số tiền bao nhiêu?**#include<stdio.h>

float tiendau(int x,float tien);

float tienduoi(int x,float tien,int i);

int main()

{

int n,i=0;

float tien=100.000;

printf("nhap so nam: ");

scanf("%d",&n);

printf("tinh theo de quy dau: %f",tiendau(n,tien));

printf("\ntinh theo de quy duoi: %f",tienduoi(n,tien,i));

}

float tiendau(int x,float tien)

{

if (x==0) return tien;

else return tiendau(x-1,tien+tien\*0.065);

}

float tienduoi(int x,float tien=100.000,int i=0)

{

if (i>=x) return tien;

else return tienduoi(x,tien+tien\*0.065,i+1);

}

**Bài 4.2: Viết các hàm đệ qui theo dạng chia để trị:**

**a) Tính S(n) = an**

#include<stdio.h>

int amun(int x,int y,int d,int c);

int main()

{

int a,n;

printf("nhap co so a: ");

scanf("%d",&a);

printf("nhap so mu n: ");

scanf("%d",&n);

printf("ket qua %d^%d la: %d",a,n,amun(a,n,1,n));

}

int amun(int x,int y, int d,int c)

{

int z=(d+c)/2;

if (d==c) return x;

else return amun(x,y,d,z)\*amun(x,y,z+1,c);

}

**b) Tính S(n)=1^2 + 2^2 + 3^2 + .. + n^2( 5 < n < 30)**

#include<stdio.h>

#include<math.h>

int tongbp(int n,int d,int c);

int main()

{

int n;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=30));

printf("tong binh phuong cua %d so tu nhien dau tien la: %d",n,tongbp(n,1,n));

}

int tongbp(int n,int d,int c)

{

if (d==c) return pow(d,2);

else

{

int g=(d+c)/2;

return tongbp(g,d,g)+tongbp(c-g,g+1,c);

}

}

**c) Tính tổng n phần tử của mảng a (a[0], a[1] ,..., a[n-1]) ( 5 < n < 100)**

#include<stdio.h>

#define max 100

int a[max];

int tongmang(int a[],int n,int d,int c);

int main()

{

int n;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=100));

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

{

printf("nhap a[%d]: ",i+1);

scanf("%d",&a[i]);

}

printf("tong cac phan tu trong mang la: %d",tongmang(a,n,0,n-1));

}

int tongmang(int a[],int n,int d,int c)

{

if(d==c) return a[d];

else

{

int g=(d+c)/2;

return tongmang(a,g-d+1,d,g)+tongmang(a,c-g,g+1,c);

}

}

**d) Đếm số lần xuất hiện của phần tử x trong mảng a (a[0], a[1] ,..., a[n-1]) ( 5 < n <**

**100)**

#include<stdio.h>

#define max 100

int a[max],dem=0;

int demx(int a[],int n,int x,int d,int c);

int main()

{

int n,x;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=100));

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

{

printf("nhap a[%d]: ",i+1);

scanf("%d",&a[i]);

}

printf("nhap so can dem: ");

scanf("%d",&x);

printf("co %d so %d trong mang",demx(a,n,x,0,n-1)),x;

}

int demx(int a[],int n,int x,int d,int c)

{

if (d==c)

if (a[d]==x) dem=dem+1;

else return dem;

else

{

int g=(d+c)/2;

demx(a,g-d+1,x,d,g);

demx(a,c-g,x,g+1,c);

}

return dem;

}

**e) Tìm phần tử có giá trị x trong mảng a (a[0], a[1] ,..., a[n-1]), ( 5 < n < 100) giả sử**

**mảng a đã được sắp xếp theo chiều tăng dần**

#include<stdio.h>

#define max 100

int a[max];

bool timi(int a[],int n,int i,int d,int c);

int main()

{

int n,i;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=100));

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

{

printf("nhap a[%d]: ",i+1);

scanf("%d",&a[i]);

}

printf("nhap so can tim: ");

scanf("%d",&i);

if (timi(a,n,i,0,n-1)==true) printf("so %d co xuat hien truong mang",i);

else printf("so %d khong xuat hien truong mang",i);

}

bool timi(int a[],int n,int i,int d,int c)

{

if (d==c)

if (a[d]==i) return true;

else return false;

else

{

int g=(d+c)/2;

if (i==a[g]) return true;

else if (i<a[g]) return timi(a,g-d+1,i,d,g);

else return timi(a,c-g,i,g+1,c);

}

}

**Bài 4.3: Viết các hàm khử đệ qui thực hiện các yêu cầu sau:**

**a) Một khách hàng gửi ngân hàng 100.000 với lãi suất 6.5%/năm. Hỏi sau n năm**

**khách hàng thu được số tiền bao nhiêu?**

#include <stdio.h>

#define MAX 100

int top=-1;

float s[MAX];

//ham stack

void ghivao(float x);

float layra();

bool trong();

bool day();

//

float tien(int n,float a);

//

int main()

{

int n;

float a;

printf("nhap so nam: ");

scanf("%d",&n);

printf("nhap so tien ban dau: ");

scanf("%f",&a);

printf("tong so tien nhan duoc sau %d nam la: %f",n,tien(n,a));

}

//

float tien(int n,float a)

{

float b=a;//tong so tien sau n nam

//dua vao stack

while(n>0)

{

ghivao(a\*0.065);

a+=a\*0.065;

n-=1;

}

//lay ra tu stack

while(!trong())

b+=layra();

return b;

}

float layra()

{

if(!trong()){

float x = s[top];

top--;

return x;

}

else

printf("trong");

}

void ghivao(float x){

if (!day()){

top++;

s[top]=x;

}

else

printf("day");

}

bool trong()

{

if (top<0)

return true;

else

return false;

}

bool day()

{

return top>=MAX;

}

**b) Đổi số nguyên dương n sang hệ nhị phân.**

#include <stdio.h>

#define MAX 100

int top=-1;

int s[MAX];

//ham stack

void ghivao(int x);

int layra();

bool trong();

bool day();

//

void nhiphan(int n);

//

int main()

{

int n;

printf("nhap n: ");

scanf("%d",&n);

printf("dang nhi phan cua %d la: ",n);

nhiphan(n);

}

//

void nhiphan(int n)

{

//dua vao stack

while(n>0)

{

ghivao(n%2);

n/=2;

}

//lay ra tu stack

while(!trong())

printf("%d",layra());

}

int layra()

{

if(!trong()){

int x = s[top];

top--;

return x;

}

else

printf("trong");

}

void ghivao(int x){

if (!day()){

top++;

s[top]=x;

}

else

printf("day");

}

bool trong()

{

if (top<0)

return true;

else

return false;

}

bool day()

{

return top>=MAX;

}

**c) Đếm số lần xuất hiện của phần tử x trong mảng a (a[0], a[1] ,..., a[n-1]) ( 5 < n <**

**100)**

#include <stdio.h>

#define MAX 100

int top=-1;

bool s[MAX];

int a[MAX];

//ham stack

void ghivao(bool x);

int layra();

bool trong();

bool day();

bool kt(int x, int y);

//

int demx(int a[],int n,int x);

//

int main()

{

int n,x;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=100));

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

{

printf("nhap a[%d]: ",i+1);

scanf("%d",&a[i]);

}

printf("nhap so can dem: ");

scanf("%d",&x);

printf("so %d xuat hien trong mang %d lan",x,demx(a,n-1,x));

}

//

int demx(int a[],int n,int x)

{

int dem=0;

//dua vao stack

while(n>=0)

{

ghivao(kt(a[n],x));

n-=1;

}

//lay ra tu stack

while(!trong())

if(layra()==true) dem+=1;

return dem;

}

int layra()

{

if(!trong()){

bool x = s[top];

top--;

return x;

}

else

printf("trong");

}

void ghivao(bool x){

if (!day()){

top++;

s[top]=x;

}

else

printf("day");

}

bool trong()

{

if (top<0)

return true;

else

return false;

}

bool day()

{

return top>=MAX;

}

bool kt(int x, int y)

{

if(x==y) return true;

else return false;

}

**d) Tìm phần tử có giá trị x trong mảng a (a[0], a[1] ,..., a[n-1]), ( 5 < n < 100) giả sử**

**mảng a đã được sắp xếp theo chiều tăng dần.**

#include <stdio.h>

#define MAX 100

int top=-1;

bool s[MAX];

int a[MAX];

//ham stack

void ghivao(bool x);

int layra();

bool trong();

bool day();

bool kt(int x, int y);

//

void timx(int a[],int n,int x,int d,int c);

//

int main()

{

int n,x;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=100));

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

{

printf("nhap a[%d]: ",i+1);

scanf("%d",&a[i]);

}

printf("nhap so can tim: ");

scanf("%d",&x);

timx(a,n,x,0,n-1);

}

//

void timx(int a[],int n,int x,int d,int c)

{

int dem=0;

int g=(d+c)/2;

//dua vao stack

if (x<=a[g])

while(g>=d)

{

ghivao(kt(a[g],x));

g--;

}

else

while(c>g)

{

ghivao(kt(a[c],x));

c--;

}

//lay ra tu stack

while(!trong())

if(layra()==true) dem+=1;

if(dem>0) printf("tim duoc %d trong mang",x);

else printf("khong tim duoc %d trong mang",x);

}

int layra()

{

if(!trong()){

bool x = s[top];

top--;

return x;

}

else

printf("trong");

}

void ghivao(bool x){

if (!day()){

top++;

s[top]=x;

}

else

printf("day");

}

bool trong()

{

if (top<0)

return true;

else

return false;

}

bool day()

{

return top>=MAX;

}

bool kt(int x, int y)

{

if(x==y) return true;

else return false;

}

**e) Tìm phần tử có giá trị lớn nhất trong mảng a(a[0], a[1] ,..., a[n-1]) ( 5 < n < 100)**

#include <stdio.h>

#define MAX 100

int top=-1;

int s[MAX];

int a[MAX];

//ham stack

void ghivao(int x);

int layra();

bool trong();

bool day();

//

int timmax(int a[],int n);

//

int main()

{

int n,x;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=5)||(n>=100));

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

{

printf("nhap a[%d]: ",i+1);

scanf("%d",&a[i]);

}

printf("phan tu lon nhat trong mang la: %d",timmax(a,n-1));

}

//

int timmax(int a[],int n)

{

int max=a[0];

//dua vao stack

while (n>0)

{

if (a[n]>=max) ghivao(a[n]);

n-=1;

}

//lay ra tu stack

while(!trong())

{

int z=layra();

if (z>max) max=z;

}

return max;

}

int layra()

{

if(!trong()){

int x = s[top];

top--;

return x;

}

else

printf("trong");

}

void ghivao(int x){

if (!day()){

top++;

s[top]=x;

}

else

printf("day");

}

bool trong()

{

if (top<0)

return true;

else

return false;

}

bool day()

{

return top>=MAX;

}

**f) Vi trùng cứ 1 giờ lại nhân đôi. Vậy sau n giờ sẽ có mấy con vi trùng nếu ban đầu**

**có 2 con?**

#include <stdio.h>

#define MAX 100

int top=-1;

int s[MAX];

//ham stack

void ghivao(int x);

int layra();

bool trong();

bool day();

//

int vitrung(int n,int a);

//

int main()

{

int n;

printf("nhap so gio: ");

scanf("%d",&n);

printf("tong so vi trung sau %d gio la: %d",n,vitrung(n,2));

}

//

int vitrung(int n,int a)

{

int virus=a;

//dua vao stack

while(n>0)

{

ghivao(a);

a=a\*2;

n-=1;

}

//lay ra tu stack

while(!trong())

virus+=layra();

return virus;

}

int layra()

{

if(!trong()){

int x = s[top];

top--;

return x;

}

else

printf("trong");

}

void ghivao(int x){

if (!day()){

top++;

s[top]=x;

}

else

printf("day");

}

bool trong()

{

if (top<0)

return true;

else

return false;

}

bool day()

{

return top>=MAX;

}

**g) Tìm giá trị Fibonacci thứ n (0 < n < 20)**

#include <stdio.h>

#define MAX 100

int top=1;

int s[MAX];

//ham stack

void ghivao(int x);

int layra();

bool trong();

bool day();

//

int Fibonacci(int n);

//

int main()

{

s[0]=1,s[1]=1;

int n,x;

do

{

printf("nhap n: ");

scanf("%d",&n);

}

while((n<=0)||(n>=20));

printf("gia tri Fibonacci thu %d la: %d",n,Fibonacci(n));

}

//

int Fibonacci(int n)

{

int a,b;

while (n>1)

{

a=layra();

b=layra();

ghivao(a);

ghivao(a+b);

n-=1;

}

return a=layra();

}

int layra()

{

if(!trong()){

int x = s[top];

top--;

return x;

}

else

printf("trong");

}

void ghivao(int x){

if (!day()){

top++;

s[top]=x;

}

else

printf("day");

}

bool trong()

{

if (top<0)

return true;

else

return false;

}

bool day()

{

return top>=MAX;

}