1.bai a mu b

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<form action="#" method="POST">

<center><table>

<tr>

<td>Nhập số a:</td>

<td><input type="text" name="soa"/></td>

</tr>

<tr>

<td>Nhập số b:</td>

<td><input type="text" name="sob"/></td>

</tr>

<center><tr>

<td></td>

<td><input type="submit" value="Tính a mũ b"/></td>

</tr></center>

</form>

</table></center>

</form>

<?php

if(!empty($\_POST)){

if(empty($\_POST['soa'])||empty($\_POST['sob'])){

echo "<center>bạn nhập thiếu rồi, nhập lại đi</center>";

}

else{

$a='1';

for($i=0;$i<$\_POST['sob'];$i++)

{

$a\*=$\_POST['soa'];

}

echo "<center> Giá trị của a mũ b:".$a."</center>";

}

}

?>

</body>

</html>

2.bài kẻ bảng

<form action="#" method = "post">

<table>

<tr>

<td colspan="2" align = "center"><h1>Form kẻ bảng</h1></td>

</tr>

<tr>

<td>số dòng</td>

<td><input type="text" name="dong"></td>

</tr>

<tr>

<td>số cột</td>

<td><input type="text" name="cot"></td>

</tr>

<tr>

<td><button type="Reset">Nhập lại</button></td>

<td><button type="Submit">Vẽ</button></td>

</tr>

</table>

</form>

<?php

if(empty($\_POST["dong"])||empty($\_POST["cot"]))

{

echo "bạn chưa nhập dòng hoặc chưa nhập cột";

}else{

echo'<table border ="1" width = "500" height = "500">';

for($i=1; $i<=$\_POST["dong"]; $i++){

echo "<tr>";

for($j=1; $j<=$\_POST["cot"]; $j++){

echo "<td>.$i.</td>";

}

}

echo "</tr>";

echo "</table>";

};

?>

Baif3:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<center><table>

<form action="#" method="POST">

Đăng Ký

<tr>

<td>Mã sinh viên:</td>

<td><input type="text" name="code"/></td>

</tr>

<tr>

<td>Họ tên:</td>

<td><input type="text" name="name"/></td>

</tr>

<tr>

<td>Ngày sinh:</td>

<td><input type="text" name="birth"/></td>

</tr>

<tr>

<td>Giới tính:</td>;

<td><input type="radio" name="gender" value="male"/>Nam

<input type="radio" name="gender" value="female"/>Nữ</td>

</tr>

<tr>

<td>Nơi sinh:</td>

<td><input type="text" name="place"/></td>

</tr>

<tr>

<td>Lớp:</td>

<td><input type="text" name="class"/></td>

</tr>

<center><tr>

<td><input type="reset" value="res"/></td>

<td><input type="submit" value="sub"/></td>

</tr></center>

</form>

</table></center>

<?php

if(!empty($\_POST)){

if(empty($\_POST['code'])||empty($\_POST['name'])||empty($\_POST['birth'])||empty($\_POST['gender'])||empty($\_POST['place'])||empty($\_POST['class'])){

echo "bạn nhập thiếu rồi";

}

else {

echo "<center><table width=200 height =200 border=0>";

echo "<tr>";

echo "<td> Mã sinh viên:</td>";

echo"<td>".$\_POST['code']."</td>";

echo "</tr>";

echo "<tr>";

echo "<td> Họ Tên:</td>";

echo"<td>". $\_POST['name'] ."</td>";

echo "</tr>";

echo "<tr>";

echo "<td> Giới tính:</td>";

echo"<td>". $\_POST['gender'] ."</td>";

echo "</tr>";

echo "<tr>";

echo "<td> nơi sinh:</td>";

echo"<td>". $\_POST['place'] ."</td>";

echo "</tr>";

echo "<tr>";

echo "<td> lớp:</td>";

echo"<td>". $\_POST['class'] ."</td>";

echo "</tr>";

echo "<tr>";

echo "<td> Ngày sinh:</td>";

echo"<td>". $\_POST['birth']. "</td>";

echo "</tr>";

echo "</table></center>";

}

}

?>

</body>

</html>

**4.ke bang 1 den 100**

<!DOCTYPE html>

<html>

<body>

<table border="1px">

<tr>

<?php

for($i = 1; $i <= 10; $i ++) {

echo "<td>";

for($j = 1; $j <= 10; $j ++) {

echo "" . ($i \* $j);

echo "<br>";

}

echo "</td>";

}

?>

</tr>

</table>

</body>

</html>

java sach

package blabla2;

import java.util.Scanner;

import java.util.ArrayList;

public class main {

ArrayList<sgk> arrSgk = new ArrayList<sgk>();

ArrayList<stk> arrStk = new ArrayList<stk>();

public void nhapsgk() {

Scanner scanner = new Scanner(System.in);

int soSgk = 1;

while (soSgk != 0) {

System.out.println("Nhap thoong tin sgk:");

sgk SGK = new sgk();

SGK.nhap();

arrSgk.add(SGK);

System.out.println("dung lai chon 0. nhap tiep bam so bat ki ");

soSgk = scanner.nextInt();

}

}

public void hienSgk() {

for (int i = 0; i < arrSgk.size(); i++) {

System.out.println("-----------------------------");

arrSgk.get(i).hien();

System.out.println("-----------------------------");

}

}

public void nhapstk() {

Scanner scanner = new Scanner(System.in);

int soStk = 1;

while (soStk != 0) {

System.out.println("Nhap thoong tin stk:");

stk STK = new stk();

STK.nhap();

arrStk.add(STK);

System.out.println("dung lai chon 0. nhap tiep bam so bat ki ");

soStk = scanner.nextInt();

}

}

public void hienStk() {

for (int i = 0; i < arrStk.size(); i++) {

System.out.println("-----------------------------");

arrStk.get(i).hien();

System.out.println("-----------------------------");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int check;

main test1 = new main();

do {

System.out.println("Chon 1: Nhap va tinh tien don sgk");

System.out.println("Chon 2: Nhap va tinh tien don stk");

System.out.println("Chon 3: Ket thuc");

System.out.println("Chon:");

check = scanner.nextInt();

switch (check) {

case 1:

test1.nhapsgk();

test1.hienSgk();

break;

case 2:

test1.nhapstk();

test1.hienStk();

;

break;

case 3:

break;

}

} while (check != 3);

}

}

package blabla2;

import java.util.Scanner;

public class stk extends sach{

public double thue,thanhTien;

public stk() {}

public stk(String tenSach, int soLuong, int giaTien,double thue) {

super(tenSach, soLuong, giaTien);

this.thue = thue;

}

public void nhap() {

Scanner scanner = new Scanner(System.in);

super.nhap();

System.out.println("Nhap Thue:");

thue = scanner.nextDouble();

}

public void hien() {

super.hien();

System.out.println("Thue:"+thue);

this.thanhTien = (super.soLuong \* super.giaTien) + this.thue;

System.out.println("thanh tien : " + this.thanhTien);

}

}

package blabla2;

import java.util.Scanner;

public class sgk extends sach {

public String loaiSach;

public int num;

public double thanhTien;

public sgk() {

super();

}

public sgk(String tenSach, int soLuong, int giaTien,String loaiSach, int num) {

super(tenSach, soLuong, giaTien);

this.loaiSach = loaiSach;

this.num = num;

}

public String tinhTrangSach(int x) {

switch(num) {

case 0:

loaiSach = "cu";

break;

case 1:

loaiSach = "moi";

break;

default:

break;

}

return loaiSach;

}

public void nhap() {

Scanner scanner = new Scanner(System.in);

super.nhap();

System.out.println("Nhap tinh trang sach (0-cu/1-moi):");

num = scanner.nextInt();

tinhTrangSach(num);

}

public void hien() {

super.hien();

System.out.println("Loai sach:"+loaiSach);

if(this.num == 0) {

this.thanhTien = (double)(super.giaTien \* super.soLuong\* 0.5);

System.out.println("thanh tien : "+ this.thanhTien);

}else if(this.num == 1) {

this.thanhTien = (double)(super.giaTien \* super.soLuong);

System.out.println("thanh tien : "+ this.thanhTien);

}

}

}

package blabla2;

import java.util.Scanner;

public class sach {

public String tenSach;

public int soLuong, giaTien;

public sach() {

}

public sach(String tenSach, int soLuong, int giaTien) {

this.tenSach = tenSach;

this.soLuong = soLuong;

this.giaTien = giaTien;

}

// khong dung get set

public void nhap() {

Scanner scanner = new Scanner(System.in);

System.out.println("Nhap ten sach:");

tenSach = scanner.nextLine();

System.out.println("Nhap so luong:");

soLuong = scanner.nextInt();

System.out.println("Nhap gia tien:");

giaTien = scanner.nextInt();

}

public void hien() {

System.out.println("Ten sach:" +this.tenSach);

System.out.println("So luong:"+this.soLuong);

System.out.println("Gia tien:"+this.giaTien);

}

}

De 4 nhap day so

#include<stdio.h>

#include<conio.h>

void nhap(int \*n, float a[100])

{

printf ("\nnhap phan tu cua day:");

scanf("%d", n);

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

{

printf("so thu %d:",i+1);

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

}

}

float min\_day(int n, float a[100]){

int i;

float min =a[0];

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

if(min>a[i]);

min=a[i];

return min;

}

float tich\_day(int n, float a[100]){

if(n==1)

return a[0];

else

return a[n-1]\*tich\_day(n-1,a);

}

main(){ int c,n;

float min\_f,a[100],tich\_f;

while(c!=4)

{

printf("\nChon 1: Nhap day so");

printf("\nChon 2: Tim phan tu nho nhat");

printf("\nChon 3: Tinh tich day");

printf("\nChon 4: Ket thuc");

printf("\nMoi ban chon: ");

scanf("%d",&c);

switch(c)

{

case 1:

nhap(&n,a);

break;

case 2:

min\_f= min\_day(n, a);

printf("\nPhan tu nho nhat: %.2f",min\_f);

break;

case 3:

tich\_f= tich\_day(n,a);

printf("\nTich day: %.2f",tich\_f);

break;

}

}

}

De 2 tim nhan vien theo ten

#include<stdio.h>

#include<conio.h>

#include<string.h>

typedef struct{

char ten[50];

int Ngaysinh;

char chucvu[50] ;

}Nhanvien;

void nhap\_ds(int \*n,Nhanvien nv[100]){

printf("\n nhap so nhan vien:");

scanf("%d", n);

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

{

printf("\nnhap thong tin nhan vien:%d", i+1);

printf("\nnhap ho ten:");

fflush(stdin);

gets(nv[i].ten);

printf("\nnhap ngay sinh:");

scanf("%d",&nv[i].Ngaysinh);

printf("\nnhap chuc vu:");

fflush(stdin);

gets(nv[i].chucvu);

}

}

void hienthi(int n, Nhanvien nv[100]){

printf("\nDANH SACH NHAN VIEN:");

printf("\nHO TEN NGAYSINH CHUCVU");

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

printf("\n%s \t%d \t%s", nv[i].ten,nv[i].Ngaysinh,nv[i].chucvu);

}

}

void Tim\_ten(int n, Nhanvien nv[100]){

int i=0;

printf("TIM SINH VIEN THEO TEN");

char name[50];

printf("nhap ten nhan vien:");

fflush(stdin);

gets(name);

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

if(strcmp(nv[i].ten,name)==0){

printf("\nDANH SACH NHAN VIEN:");

printf("\nHO TEN NGAYSINH CHUCVU");

printf("\n%s \t%d \t%s", nv[i].ten,nv[i].Ngaysinh,nv[i].chucvu);

break;

}

}

if(i==n){

printf("khong tim thay");

}

}

main(){

int chon, n;

Nhanvien nv[100];

while(chon!=4){

printf("\n1. Chon 1 nhap danh sach nhan vien.");

printf("\n2. Chon 2 in danh sach.");

printf("\n3. chon 3 tim kiem theo ten");

printf("\n4.Chon 4 ket thuc\n");

scanf("%d:", &chon);

switch(chon){

case 1:

nhap\_ds(&n ,nv);

break;

case 2:

hienthi(n,nv);

break;

case 3:

Tim\_ten(n,nv);

break;

}

}

}

Cau 10 java

package blabla1;

import java.util.Scanner;

public class cau10 {

int n;

public static int a[];

Scanner scanner = new Scanner(System.in);

public void nhap() {

System.out.println("Nhap so phan tu cua mang:");

n = scanner.nextInt();

a = new int[n];

for (int i = 0; i<a.length;i++) {

System.out.println("Phan tu so:" + (i+1));

a[i] = scanner.nextInt();

}

System.out.println("Mang vua nhap la:");

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

System.out.println(a[i] + "");

}

System.out.println();

}

public int max(int a[], int n) {

if (n==0) {

return -1;

}

if (n==1) {

return a[0];

}

else {

if (a[n-1] < max(a,n-1)) {

return max(a,n-1);

}

else {

return a[n-1];

}

}

}

public void tinhTong() {

int tong = 0;

for (int i=0; i<a.length;i++) {

tong = tong +a[i];

}

System.out.println("tong mang vua nhap la:" +tong);

}

public static void main(String[] args) {

int check = 0, n;

Scanner sc = new Scanner(System.in);

cau10 cau10 = new cau10();

do {

System.out.println("Chon 1: Nhap day so");

System.out.println("Chon 2: Tim phan tu lon nhat");

System.out.println("Chon 3: Tinh tong day");

System.out.println("Chon 4: Ket thuc");

System.out.println("Chon:");

check = sc.nextInt();

switch(check ) {

case 1:

cau10.nhap();

break;

case 2:

System.out.println("Gia tri lon nhat cua mang la:"+ cau10.max(a,a.length));

break;

case 3:

cau10.tinhTong();

break;

}

}while (check!=4);

}

}

**1.Phân biệt hệ cô lập và hệ phân tán**

Hệ cô lập  
– Là một node tính toán riêng biệt (1 PC, mobile, …)  
– Chương trình quản lý tài nguyên và truy xuất bộ nhớ trực tiếp  
– Các kỹ thuật lập trình trong các phần trước viết cho hệ cô lập  
• Hệ phân tán:  
– Tập hợp các node tính riêng rẽ có thể làm việc độc lập và liên  
kết để giải quyết công việc  
– Trong hệ phân tán, mỗi bộ xử lý nói chung có chương trình làm  
việc riêng bán độc lập, vì lợi ích chung nên cần phối hợp hành  
động với nhau  
– Chương trình trong môi trường phân tán gồm nhiều phần, chạy  
trên các node khác nhau  
– Chương trình trên mỗi node không thể trực tiếp truy xuất bộ nhớ  
và tài nguyên của node khác mà phải gián tiếp và từ sxa  
– Hệ phân tán phổ biến nhất = Mạng máy tính

1. Mô tả quá trình thực hiện và tổ chức các thành phần bên trong bộ nhớ của một hàm đệ quy.

phương thức là tập hợp các lệnh với tham số truyền vào để máy tính thao tác lệnh theo ý muốn của người viết, đệ quy xảy ra khi người viết các phương thức tự gọi (hoạc định nghĩa lại) chính nó.

 Đề bài: Tính lũy tiến từ 0 đến n

public int sum(int n) {

if (n >= 1) {

return sum(n - 1) + n;

}

return n;

}

Giải thích:

* Bạn truyền một tham số n vào phương thức sum(), lệnh trong phương thức sum sẽ trả về tham số n bạn truyền vào khi chạy hết chương trình “return n”.
* Để đến được bước đó, chương trình sẽ chạy qua các lệnh điều kiện “if(n>=1)” để định nghĩa lại phương thức sum một lần nữa “sum(n-1) + n”, phương thức mới sẽ khiến giá trị n sẽ thay đổi theo từng vòng của điều kiện cho đến khi không còn thỏa mãn điều kiện được cho.
* Khi chương trình “return n” thì n chính là giá trị đã được tính ở phương thức ta đặt điều kiện bên trên.

Như vậy, hai yếu tố cần để tiến hành một phương thức đệ quy là:

* **Có điều kiện dừng**: Xác định quy luật của phương thức và tìm giá trị cụ thể khi thỏa mãn một điều kiện nhất định, ở bước này vẫn chưa có phương thức đệ quy nào được gọi.
* **Phương thức đệ quy**: Phương thức đệ quy sẽ gọi lại chính nó cho đến khi nó trả về điều kiện dừng ở bước 1.

​ mỗi lần bạn sử dụng đệ quy, chương trình chạy một vòng và bộ nhớ Stack sẽ được chồng thêm một lớp dữ liệu, tình trạng lãng phí bộ nhớ rất dễ xảy ra nếu bạn không phân tích kỹ các vòng chạy đệ quy để có tính toán hợp lý. Vấn đề trên có thể giải quyết bằng cách “tối ưu hóa đòn bẩy đệ quy đuôi”.

1. **Bản chất quá trình chuyển điều khiển giữa chương trình chính và chương trình con. Tại sao trong chương trình sử dụng nhiều chương trình con lại làm tăng thời gian thực hiện.**

* Chuyển là chuyển thanh ghi lệnh đến vùng code của chương trình con. Mất thời gian tìm kiếm. Sau khi kết thúc chương trình con thì bạn phải giải phón stack dữ liệu của nó khôi phục trạng thái của chương trình gọi nó Mất thời gian.

**7.Giải thích cách tính miền giá trị của các kiểu dữ liệu. Mô tả cách tổ chức của mảng trong bộ nhớ. Trong C một mảng có tối đa bao nhiêu phần tử.**

Ví dụ về char: 1 bit trong lập trình sẽ biểu diễn được 2 giá trị: 0 hoặc 1. Tức là biểu diễn được 21 giá trị.

Vậy 2 bits sẽ biểu diễn được 4 giá trị: 00, 01, 10 và 11. Tức là biểu diễn được 22 giá trị.

Như vậy, 1 byte = 8 bits sẽ biểu diễn được 28 giá trị khác nhau.

Mà máy tính lưu giá trị mã bit 0 và 1. Tức dạng nhị phân. Do đó 00 = 0, 01 = 1, 10 = 2, 11 = 3 trong cơ số 10. Như vậy, giá trị lớn nhất mà 1 byte có thể biểu diễn là 11111111 = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255

- Đó là lý do kiểu unsigned char có phạm vi giá trị từ 0 đến 255. Còn kiểu char do có cả miền âm nên sẽ bị chia đôi(-128 đến 127). Phần dương bị hụt 1 giá trị do cần biểu diễn cả số 0

* <đọc thêm cho vào thì cho> Để tìm giới hạn của kiểu [dữ liệu](https://en.wikipedia.org/wiki/C_data_types) nguyên thủy, chúng ta cần < **limit.h** >. Trong không dấu, giá trị nhỏ nhất không có vì trong không dấu, các giới hạn bắt đầu bằng 0 có thể được nhớ dễ dàng. Tệp hệ thống limit.h có sẵn trong trình biên dịch C-Compliant chứa các giới hạn trên và dưới của các kiểu số nguyên. Người dùng có thể bao gồm trước main ()
* **Ví dụ**
* printf ("\ nGiá trị nhỏ nhất của CHAR đã ký là =% d \ n", SCHAR\_MIN);
* printf ("Giá trị lớn nhất của CHAR đã ký là =% d \ n", SCHAR\_MAX);
* printf ("Giá trị nhỏ nhất của CHAR là =% d \ n", CHAR\_MIN);
* printf ("Giá trị lớn nhất của CHAR là =% d \ n", CHAR\_MAX);
* printf ("Giá trị lớn nhất của CHAR Chưa ký là =% u \ n", UCHAR\_MAX);
* printf ("\ nGiá trị nhỏ nhất của Ngắn đã ký là =% d \ n", SHRT\_MIN);
* printf ("Giá trị lớn nhất của Ngắn đã ký là =% d \ n", SHRT\_MAX);
* printf ("Giá trị lớn nhất của Ngắn chưa được ký là =% u \ n", USHRT\_MAX);
* printf ("\ nGiá trị nhỏ nhất của INT đã ký là =% d \ n", INT\_MIN);
* printf ("Giá trị lớn nhất của INT đã ký là =% d \ n", INT\_MAX);
* printf ("Giá trị lớn nhất của INT không dấu là =% u \ n", UINT\_MAX);
* printf ("\ nGiá trị nhỏ nhất của LONG đã ký là =% ld \ n", LONG\_MIN);
* printf ("Giá trị lớn nhất của LONG đã ký là =% ld \ n", LONG\_MAX);
* printf ("Giá trị lớn nhất của DÀI không dấu là =% lu \ n", ULONG\_MAX);
* Mangr là  một [cấu trúc dữ liệu](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u) bao gồm một nhóm các phần tử giá trị hoặc [biến](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_s%E1%BB%91), mỗi phần tử được xác định ít nhất bằng một *chỉ số* (index) hoặc *khóa* (key). Mảng được lưu theo cách có thể tính được vị trí của các phần tử từ giá trị của một [tuple](https://vi.wikipedia.org/w/index.php?title=Tuple&action=edit&redlink=1) chỉ số bằng một [biểu thức](https://vi.wikipedia.org/w/index.php?title=Bi%E1%BB%83u_th%E1%BB%A9c&action=edit&redlink=1) toán học.
* khi làm việc với mảng, chúng ta có thể truy cập trực tiếp tới phần tử nào đó thông qua chỉ mục của nó (ví dụ: arr[5] dùng để truy cập đến phần tử thứ 6). Trở lại khái niệm của mảng thì các phần tử được lưu trong các ô nhớ liên tiếp nhau hoặc lưu theo cách có thể tính được vị trí của các phần tử bằng một biểu thức toán học. Và chỉ mục chính là tham số chính để xác định được vị trí của phần tử trong bộ nhớ.
  1. Bản chất biến là gì; địa chỉ ô nhớ là gì; con trỏ là gì; con trỏ hàm là gì. Ý nghĩa sử dụng của con trỏ hàm; cho ví dụ minh họa.

- Bản chất của Biến: là vùng nhớ dùng để lưu trữ các giá trị của chương trình. Mỗi biến gắn liền với một kiểu dữ liệu và một định danh duy nhất gọi là tên biến.

- Con trỏ hàm là một biến lưu trữ địa chỉ của một hàm, thông qua biến đó, ta có thể gọi hàm mà nó trỏ tới. Điều này rất thuận tiện khi bạn muốn định nghĩa các chức năng khác nhau cho một nhóm các đối tượng khá giống nhau.

*Con trỏ đơn giản là một biến kiểu số nguyên dùng để chứa địa chỉ.*

**Các con trỏ hàm trong C có thể được sử dụng để thực hiện lập trình hướng đối tượng trong C**