Bài 3: Hãy viết chương trình có sử dụng hàm để tính lãi suất tiết kiệm. Bạn có một khoản tiền P để gửi vào quỹ tiết kiệm, mỗi tháng lãi I phần trăm. Số tiền sau n tháng gửi sẽ là F=P(1 +I/100)^n F chính là số tiền cả vốn lẫn lãi nhận được sau n tháng. Tỷ số F/P thường được tính và in ra thành bảng để nhân viên ngân hàng hoặc người gửi tra cứu. Tỷ số này được in ra thành bảng theo tỉ lệ lãi và theo số tháng. Hãy lập bảng sau:

1. Thuật toán :

* Bước 1:Nhập số tiền gửi P
* Bước 2: viết hàm tính lãi F với công thức P \* pow(1 + I / 100, n) Với: P là số tiền gửi I là mức lãi(5%,6%,..)
* Bước 3:In bảng tính lãi Với các mức lãi và tỷ số tiễn lãi/tiền gốc theo các tháng.(F/P)

1. Chương trình nguồn :

#include <stdio.h>

#include <conio.h>

#include <math.h>

float tinhLai(float P, float I, int n)

{

return P \* -pow(1 + I / 100, n);

}

void inBangTinhLai(float P)

{

int i,n,lai[] = {5, 6, 7, 8, 9, 10};

int soLai = sizeof(lai) / sizeof(lai[0]);

printf("n");

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

{

printf(" ");

printf("%d%%", lai[i]);

}

printf("\n");

for (n = 1; n <= 10; n++)

{

printf("%-6d", n);

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

{

float F = tinhLai(P, lai[i], n) / P;

printf("%10.5f", F);

printf(" ");

}

printf("\n");

}}

int main()

{

float P;

printf("nhap so tien:");

scanf("%f", &P);

inBangTinhLai(P);

getch();

}

1. Kết quả :

A screenshot of a computer screen

Description automatically generated

Bài 34. Viết chương trình nhập vào chuỗi số nguyên dương a và kiểm tra a có phải là chuỗi số tự đối xứng không? Ví dụ: 5, 232, 5775 tự đối xứng, nhưng 2342 thì không tự đối xứng. 01 là không đối xứng

1. Thuật toán :

* Bước 1 : Nhập chuối số từ bán phím.
* Bước 2 : So sánh hai kí từ đầu (a[i]) và cuối (a[j]). Trong đó i bắt đàu từ 0 , j bắt đầu từ độ dài của mảng -1.
* Bước 3 : Nếu trong quá trình kiểm tra có phần nào a[i]!=a[j] thì dừng lại và kết luận chuối số không tự đói xứng. Còn không thì kết luận chuỗi số tự đối xứng.

1. Chương trình nguồn :

#include<stdio.h>

#include<string.h>

int kiemtradoixung(char a[]);

int main(){

char a[100];

gets(a);

int t=kiemtradoixung(a);

if(t==0){printf("Khong doi xung");

}

else printf("doi xung");

return 0;

}

int kiemtradoixung(char a[]){

int i=0,j=strlen(a)-1;

while (i<=j){

if(a[i]!=a[j]){

return 0;

}

i++;j--;

}

return 1;

}

1. Kết quả :

A black screen with white text

Description automatically generated

A black screen with white text

Description automatically generated

Bài 37. (4 điểm): Cho số tự nhiên N (N<=7000000). Phân tích N thành tổng các số chính phương nhỏ dần sao cho số các số hạng là ít nhất. Ví dụ: 30 =5^2+1^2+2^2.

1. Thuật toán :

* Bước 1 : Nhập số nguyên N.
* Bước 2 : Tìm căn bậc 2 của N (Sử dụng sqrt(N)).
* Bước 3 : Bình phương kết quả vừa rồi và nó trở thành một số chính phương bé hơn hoặc bằng N.
* Bước 4 : Lấy N trừ cho số chính phương đó và lặp lại quá trình trên cho đén khi N=0.
* Bước 5 : in các số chính phương vừa tính ở trên ra mà hình.

1. Chương trình nguồn :

#include<stdio.h>

#include<math.h>

void phantichsochinhphuong(int x);

int main(){

int n;

printf("Nhap so N: ");

scanf("%d",&n);

phantichsochinhphuong(n);

return 0;

}

void phantichsochinhphuong(int x){

printf("\nCac so chinh phuong la: ");

while(x>0){

int y=(int)sqrt(x);

int temp=y\*y;

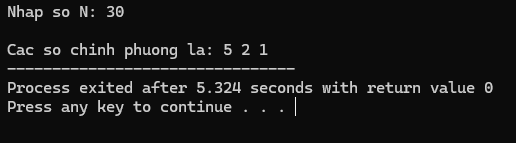
x-=temp;

printf("%d ",y);

}

}

1. Kết quả :



Bài 63:Viết chương trình nhập mảng 1 chiều A có N phần tử có sử dụng hàm(với N nhập từ bàn phím). a, Xuất các phần tử đã nhập ra màn hình

b, Xuất ra màn hình các chính phương của mảng và tính tổng các số đó?

1. Thuật toán :

* Bước 1 : Nhập số phần tử N(N>0)
* Bước 2 : Hàm nhập mảng :
* Cho hàm chạy từ i=0 dến N-1.
* Nhập từng phần tử A[i].
* Bước 3 hàm xuất mảng :
* Cho hàm chạy từ i=0 dến N-1.
* Xuất từng phần tử A[i].
* Bước 4 : Kiểm tra số chính phương :
* Khai báo biến s=0
* Với mỗi phần tử A[i], kiểm tra xem A[i] có phải là số chính phương
* Khai báo biến temp bằng căn bặc 2 của A[i] và lấy phần nguyên.
* Kiểm tra nếu temp\*temp bằng A[i] thì là số chính phương, s =s+A[i].
* Lắp lại với các phần tử khác.
* Bước 5 :
* In các số chính phương vừa kiểm tra ở trên và in s ra màn hình.

1. Chương trình nguồn :

#include<stdio.h>

void nhap(int a[], int n);

void xuat(int a[], int n);

void xuatscp(int a[], int n);

int main(){

int n;printf("Nhap so phan tu : ");

scanf("%d",&n);

int a[n];

printf("Nhap cac phan tu : ");

nhap(a,n);

printf("Cac phan tu da nhap la : ");

xuat(a,n);

printf("\nCac so chinh phuong cua mang la : ");

xuatscp(a,n);

return 0;

}

void nhap(int a[], int n){

int i=0;

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

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

}

}

void xuat(int a[], int n){

int i=0;

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

printf("%d ",a[i]);

}

}

void xuatscp(int a[], int n){

int i=0,s=0;

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

int temp =(int)sqrt(a[i]);

if(temp\*temp==a[i]){

printf("%d ",a[i]);

s+=a[i];

}

}

printf("\nTong cac so chinh phuong la %d",s);

}

1. Kết quả :

A screenshot of a computer

Description automatically generated

Bài 66: Viết chương trình có sử dụng hàm .

a, Viết chương trình nhập mảng một chiều A. Xuất mảng A ra màn hình

b, Xóa các phần âm trong mảng A.Xuất mảng A sau khi xóa các phần tử âm ra màn hình.

1. Thuật toán :

* Bước 1 : Nhập số phần tử N(N>0)
* Bước 2 : Hàm nhập mảng :
* Cho hàm chạy từ i=0 dến N-1.
* Nhập từng phần tử A[i].
* Bước 3 : Hàm xuất mảng :
* Cho hàm chạy từ i=0 dến N-1.
* Xuất từng phần tử A[i].
* Bước 4 : Kiểm tra xem A[i] có âm không. Nếu âm thì dịch các phần tử phái sau lên 1 vị trí. Tiếp tục cho đến phần tử cuối .
* Bước 5 : In mảng ra màn hình.

1. Chương trình nguồn :

#include<stdio.h>

void nhap(int a[], int n);

void xuat(int a[], int n);

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

int main(){

int n;printf("Nhap so phan tu : ");

scanf("%d",&n);

int a[n];

printf("Nhap cac phan tu : ");

nhap(a,n);

printf("Cac phan tu da nhap la : ");

xuat(a,n);

n=xoaam(a,n);

printf("\nCac phan tu sau khi chinh sua la : ");

xuat(a,n);

return 0;

}

void nhap(int a[], int n){

int i=0;

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

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

}

}

void xuat(int a[], int n){

int i=0;

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

printf("%d ",a[i]);

}

}

int xoaam(int a[], int n){

int i=0;

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

if(a[i]<0){

int j=i;

for(j;j<n;j++){

a[j]=a[j+1];

}

n--;

}

}

return n;

}

1. Kết quả :

A black screen with white text

Description automatically generated