**Mục Lục**

[Tam giác Pascal 3](#_Toc48860861)

[3 hàm trong thư viện string.h thuộc về C 4](#_Toc48860862)

[Hàm *con* Phân tích số nguyên tố 5](#_Toc48860863)

[Kiểm tra số chính phương 5](#_Toc48860864)

[Bội chung nhỏ nhất 6](#_Toc48860865)

[Ước chung lớn nhất (5 cách - update) 7](#_Toc48860866)

[Đổi số thập phân sang nhị phân 9](#_Toc48860867)

[Đổi số nhị phân sang thập phân 10](#_Toc48860868)

[Căn bậc 2 của 1 số (nếu đề ko cho dùng sqrt) 11](#_Toc48860869)

[Dãy số Fibonacci (update) 11](#_Toc48860870)

[Tính giai thừa của 1 số 12](#_Toc48860871)

[Sắp xếp tăng dần trong Mảng 12](#_Toc48860872)

[Ma trận chuyển vị 13](#_Toc48860873)

[Tổng và tích của 2 ma trận 13](#_Toc48860874)

[Kiểm tra tam giác 14](#_Toc48860875)

[Chuyển chữ thường sang chữ in hoa 15](#_Toc48860876)

[Chuyễn chữ hoa sang chữ thường 15](#_Toc48860877)

[Hàm Strlen Nhân Tạo 16](#_Toc48860878)

[Hàm Strcpy Nhân Tạo (update) 16](#_Toc48860879)

[Hàm Strcmp Nhân Tạo 16](#_Toc48860880)

[Kiểm tra ngày tháng năm có tồn tại hay ko? 17](#_Toc48860881)

[Kiểm tra bộ 3 Pytago 19](#_Toc48860882)

[Kiểm tra tuổi con giáp 19](#_Toc48860883)

[Giải PT bậc 2 ax2+bx+c=0 21](#_Toc48860884)

[Kiểm tra số đó có phải là luỹ thừa ko? 22](#_Toc48860885)

[Tìm số nguyên tố ở vị trí n 23](#_Toc48860886)

[Giai thừa nhỏ nhất 24](#_Toc48860887)

[In tam giác theo ký tự \* 24](#_Toc48860888)

[Tìm chuỗi con dài nhất mà không có kí tự lặp lại 25](#_Toc48860889)

[Nhập nhiều chuỗi và sắp xếp chúng theo chiều tăng dần 26](#_Toc48860890)

[Tìm từ ngắn nhất và từ dài nhất trong 1 chuỗi dài 27](#_Toc48860891)

[Ma trận đối xứng 28](#_Toc48860892)

[Liệt kê tích số nguyên tố từ 2 đến n với n nhập từ bàn phím 28](#_Toc48860893)

[Chuyển hoa thành thường và thường thành hoa trong 1 chuỗi 30](#_Toc48860894)

# Tam giác Pascal

int tamgiacPascal(int k, int n) {

if (k == 0 || k == n) return 1;

if (k == 1) return n;

return tamgiacPascal(k - 1, n - 1) + tamgiacPascal(k, n - 1);

}

int main() {

int n;

scanf("%d", &n);

printf("---------PASCAl---------\n");

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

for (int j = n; j > i; j--)

printf(“ “);

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

printf("%d ", tamgiacPascal(j, i));

}

printf("\n");

}

}

# 3 hàm trong thư viện string.h thuộc về C

int strlen\_outinlucde(char s[])

{

int z = 0;

while (s[z] != '\0')

z++;

return z;

}

void strcpy\_outinclude(char X[], char Y[])

{

int m = 0;

do

{

X[m] = Y[m];

m++;

} while (Y[m] != '\0');

}

int strcmp\_outinclude(char q[], char p[])

{

int u = 0;

do

{

if (q[u] > p[u])

return 1;

else if (q[u] < p[u])

return -1;

u++;

} while (q[u] != '\0' || p[u] != '\0');

return 0;

}

# Hàm *con* Phân tích số nguyên tố

void phanTichSoNguyen(int n) {

    int i = 2;

    int dem = 0;

    int a[100];

    // phan tich

    while (n > 1) {

        if (n % i == 0) {

            n = n / i;

            a[dem++] = i;

        } else {

            i++;

        }

    }// neu dem = 0 thi n la nguyen to

    if (dem == 0) {

        a[dem++] = n;

    }

    // in ket qua ra man hinh

    for (i = 0; i < dem - 1; i++) {

        printf("%d x ", a[i]);

    }

    printf("%d", a[dem - 1]);

}

# Kiểm tra số chính phương

int scp(int n)

{

int i = 0;

while (i \* i <= n) {

if (i \* i == n) {

printf("%d la so chinh phuong!\n", n);

return 0;

}

++i;

}

printf("%d khong phai so chinh phuong!\n", n);

}

# Bội chung nhỏ nhất

int bcnn(int a, int b)

{

int max, step, lcm = 0;

if (a > b)

max = step = a;

else

max = step = b;

while (1) {

if (max % a == 0 && max % b == 0) {

lcm = max;

break;

}

max += step;

}

return lcm;

}

# Ước chung lớn nhất (5 cách - update)

int ucln1(int a, int b)

{

if (a < b)

{

int temp = b;

b = a;

a = temp;

}

if (a == 0 || b == 0)

return a + b;

while (a != b)

{

if (a > b)a -= b;

else b -= a;

}

return a;

}

int ucln2(int a, int b)

{

if (a < b)

{

int temp = b;

b = a;

a = temp;

}

if (a == 0 || b == 0)

return a + b;

int r = a % b;

while (r != 0)

{

a = b;

b = r;

r = a % b;

}

return b;

}

int ucln3(int a, int b)

{

while (a \* b != 0) {

if (a > b) {

a %= b;

}

else {

b %= a;

}

}

return a + b;

}

int ucln5(int a, int b) {

if (b == 0) return a;

return ucln5(b, a % b);

}

int ucln4(int a, int b)

{

int tmp;

while (b != 0) {

tmp = a % b;

a = b;

b = tmp;

}

return a;

}

# Đổi số thập phân sang nhị phân

#include<math.h> //them thu vien

long long Dec2Bin(int thathinh)

{

long long yeucc = 0;

int p = 0;

while (thathinh > 0)

{

yeucc += (thathinh % 2) \* pow(10, p);

++p;

thathinh /= 2;

}

return yeucc;

}

# Đổi số nhị phân sang thập phân

#include<math.h> //them thu vien

int BinToDec(long long yeucc)

{

int p = 0;

int mlem = 0;

while (yeucc > 0)

{

mlem += (yeucc % 10) \* pow(2, p);

++p;

yeucc /= 10;

}

return mlem;

}

# Căn bậc 2 của 1 số (nếu đề ko cho dùng sqrt)

#define CRUSH 0.0001f

double mySqrt(int lozz)

{

double ciu = 1.0f;

while (fabs(ciu \* ciu - lozz) / lozz >= CRUSH)

ciu = (lozz / ciu - ciu) / 2 + ciu;

return ciu;

}

# Dãy số Fibonacci (update)

int fibonacci(int n) {

int f0 = 0;

int f1 = 1;

int fn = 1;

int i;

if (n < 0) {

return -1;

}

else if (n == 0 || n == 1) {

return n;

}

else {

for (i = 2; i < n; i++) {

f0 = f1;

f1 = fn;

fn = f0 + f1;

}

}

return fn;

}

int F(int n)

{

if (n == 0) return 0;

else if (n == 1) return 1;

return F(n - 1) + F(n - 2);

}

# Tính giai thừa của 1 số

void Giai\_Thua(long int& n)

{

int giai\_thua = 1;

printf("Nhap giai thua: "); scanf\_s("%d", &n);

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

giai\_thua \*= i;

printf("Ket qua giai thua cua %d la %d", n, giai\_thua);

}

# Sắp xếp tăng dần trong Mảng

void sapxepgiam(int a[], int n)

{

for (int i = 0; i < n - 1; i++)

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

{

if (a[i] < a[j])

{//neu tang thi doi thanh dau “>”

int temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

printf("\n mang sap xep giam la: ");

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

{

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

}

}

# Ma trận chuyển vị

void mxchuyenvi(int arr[][100], int size)

{

int temp;

for (int i = 1; i < size; i++) {

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

temp = arr[i][j];

arr[i][j] = arr[j][i];

arr[j][i] = temp;

}

}

}

# Tổng và tích của 2 ma trận

//Tong hai ma tran A va B luu vao trong ma tran C  
void Tong(int A[max][max], int B[max][max], int C[max][max], int n){  
   for(int i = 0; i<n ; i++)  
   for(int j = 0; j<n ; j++)  
      C[i][j] = A[i][j]+B[i][j];  
}  
//Tich hai ma tran A va B luu vao trong ma tran C  
void Tich(int A[max][max], int B[max][max], int C[max][max], int n)  
{  
   for(int i = 0; i<n ; i++)  
   for(int k = 0; k<n ; k++)  
   {  
      C[i][k] = 0;  
      for(int j = 0; j<n ; j++)  
         C[i][k] = C[i][k] + A[i][j]\*B[j][k];  
   }  
}

# Kiểm tra tam giác

void main()

{

int a, b, c;

printf( "Nhap vao canh a: ");

scanf("%d", &a);

printf( "Nhap vao canh b: ");

scanf("%d",&b);

printf( "Nhap vao canh c: ");

scanf("%d", &c);

if (a < b + c && b < a + c && c < a + b) {

if (a \* a == b \* b + c \* c || b \* b == a \* a + c \* c || c \* c == a \* a + b \* b)

printf( "Day la tam giac vuong");

else if (a == b == c)

printf( "Day la tam giac deu");

else if (a == b || a == c || b == c)

printf( "Day la tam giac can");

else if (a \* a > b \* b + c \* c || b \* b > a \* a + c \* c || c \* c > a \* a + b \* b)

printf( "Day la tam giac tu");

else

printf( "Day la tam giac nhon");

}

else

printf( "Ba canh a, b, c khong phai la ba canh cua mot tam giac");

}

# Chuyển chữ thường sang chữ in hoa

char strupr\_outinlucde(char arr[])

{

for (int i = 0; i <= strlen(arr); i++) {

if (arr[i] >= 97 && arr[i] <= 122)

arr[i] = arr[i] - 32;

}

return puts(arr);

}

# Chuyễn chữ hoa sang chữ thường

char strlwr\_outinclude(char arr[])

{

int i;

for (i = 0; i < strlen(arr); i++)

{

if (arr[i] >= 'A' && arr[i] <= 'Z')

arr[i] = arr[i] + 32;

}return puts(arr);

}

# Hàm Strlen Nhân Tạo

int strlen\_outinlucde(char s[])

{

int z = 0;

while (s[z] != '\0')

z++;

return z;

}

# Hàm Strcpy Nhân Tạo (update)

void strcpy\_outinclude(char X[], char Y[])

{

int m = 0;

do

{

X[m] = Y[m];

m++;

} while (Y[m] != '\0');

}

void strcpy\_pro(char\* dest, char\* src) {

while (\*dest++ = \*src++);

}

# Hàm Strcmp Nhân Tạo

int strcmp\_outinclude(char q[], char p[])

{// hàm so sánh 2 chuỗi

int u = 0;

do

{

if (q[u] > p[u])

return 1;

else if (q[u] < p[u])

return -1;

u++;

} while (q[u] != '\0' || p[u] != '\0');

return 0;

}

# Kiểm tra ngày tháng năm có tồn tại hay ko?

void kiemtra\_date(int ngay, int thang, int nam)

{

int ngaymax;

if (ngay < 0 || 12 < thang > 0 || nam < 0 || ngay < 0 || ngay >31)

printf("khong hop le");

else {

switch (thang) {

case 1:case 3:case 5:case 7:case 8:case 10:case 12:

ngaymax = 31;

break;

case 2:

{

if (nam % 4 == 0)

ngaymax = 29;

else

ngaymax = 28;

break;

} case 4: case 6: case 9: case 11:

ngaymax = 30;

break;

}

if (ngay < ngaymax)

printf("hop le");

else

printf("khong hop le");

}

}

## Kiểm tra bộ 3 Pytago

int main()

{

int a, b, c, d;

scanf("%d%d%d", &a, &b, &c);

bool kg = (c \* c) == (b \* b + a \* a);

printf("%d", kg);

}

## Kiểm tra tuổi con giáp

int main()

{

int a;

scanf("%d", &a);

if (a < 0) {

if ((a % 12 == 0)) { printf( "DAU");}

if ((a % 12 == -1)) { printf( "THAN");}

if ((a % 12 == -2)) { printf( "MUI");}

if ((a % 12 == -3)) { printf( "NGO");}

if ((a % 12 == -4)) { printf( "TY.");}

if ((a % 12 == -5)) { printf( "THIN");}

if ((a % 12 == -6)) { printf( "MEO");}

if ((a % 12 == -7)) { printf( "DAN");}

if ((a % 12 == -8)) { printf( "SUU");}

if ((a % 12 == -9)) { printf( "TY'");}

if ((a % 12 == -10)) { printf( "HOI");}

if ((a % 12 == -11)) { printf( "TUAT");}

}

else {

if ((a % 12 == 0)) { printf( "THAN");}

if ((a % 12 == 1)) { printf( "DAU");}

if ((a % 12 == 2)) { printf( "TUAT");}

if ((a % 12 == 3)) { printf( "HOI");}

if ((a % 12 == 4)) { printf( "TY'");}

if ((a % 12 == 5)) { printf( "SUU");}

if ((a % 12 == 6)) { printf( "DAN");}

if ((a % 12 == 7)) { printf( "MEO");}

if ((a % 12 == 8)) { printf( "THIN");}

if ((a % 12 == 9)) { printf( "TY.");}

if ((a % 12 == 10)) { printf( "NGO");}

if ((a % 12 == 11)) { printf( "MUI");}

}

}

## Giải PT bậc 2 ax2+bx+c=0

int main() {

double a, b, c, dt, x, y, z, t;

scanf("%d%d%d", &a, &b, &c);

dt = b \* b - 4 \* a \* c;

if (a != 0) {

if (dt == 0) {

x = ((-b / (2 \* a)) == -0 ? 0 : (-b / (2 \* a)));

printf("{%d}", x);

}

else if (dt > 0) {

y = (min(((-b + sqrt(dt)) / (2 \* a)),

((-b - sqrt(dt)) / (2 \* a))) == -0 ? 0 : (min(((-b + sqrt(dt)) / (2 \* a)),

((-b - sqrt(dt)) / (2 \* a)))));

z = (max(((-b - sqrt(dt)) / (2 \* a)),

((-b + sqrt(dt)) / (2 \* a))) == -0 ? 0 : (max(((-b + sqrt(dt)) / (2 \* a)),

((-b - sqrt(dt)) / (2 \* a)))));

printf("{%d, %d}", y, z);

}

else { printf("Ø"); }

}

else {

if (b == 0 & c == 0) { printf("ℝ"); }

else if (b == 0 & c != 0) { printf("Ø"); }

else {

t = ((-c / b) == -0 ? 0 : (-c / b));

printf("{%d}", t);

}

}

}

## Kiểm tra số đó có phải là luỹ thừa ko?

int main() { // bài chưa hoàn thiện, chưa xong

int mang[255];

mang[0] = 1;

int a = 1;

bool tf;

tf = false;

int t = 1;

while (mang[a - 1] != 0) {

scanf("%d",mang[a]);

a = a + 1;

}

for (int b = 1; b < a - 1; b++) {

if (mang[b] == 1) { cout << 1 << endl; continue; }

for (int z = 2; z <= mang[b] + 1; z++) {

if (tf == true) { break; }

t = z;

for (int d = 1; d <= mang[b] + 1; d++) {

t = t \* z;

if (t == mang[b]) {

cout << 1 << endl;

tf = true;

break;

}else if (t > mang[b]) break;

}

}

if (tf == false) cout << 0 << endl;

else tf = false;

}

}

## Tìm số nguyên tố ở vị trí n

unsigned long Prime(unsigned long n)

{

unsigned long i;

if (n < 2) return 0;

if (n == 2 || n == 3) return 1;

if (n % 2 == 0 || n % 3 == 0) return 0;

for (i = 5; i <= sqrt(n); i += 6)

if (n % i == 0 || n % (i + 2) == 0)

return 0;

return 1;

}

int main()

{

unsigned long i = 1, dem = 0, n;

scanf("%d", &n);

while (dem != n)

{

i++;

if (Prime(i) == 1)

dem++;

}

printf("%d\n", i);

}

## Giai thừa nhỏ nhất

//in số kế tiếp là giai thừa

// vd: 17 thì in ra 24 = 4!

int leastFactorial(int n) {

int gt = 1, i = 1;

while (gt < n) {

gt = gt \* i;

i++;

}

return gt;

}

## In tam giác theo ký tự \*

int main()

{

int n;

scanf("%d", &n);

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

{

// in ky tu khoang trang

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

cout << " ";

// in ky tu ngoi sao

for (int j = 1; j <= (2 \* i - 1); j++)

cout << "\* ";

// xuong dong ke tiep

printf("\n\n");

}

}

## Tìm chuỗi con dài nhất mà không có kí tự lặp lại

void main()

{

char s[80], mlem[80];

int n, i, j, k, t = 0, length = 0, longest = 0, repeated = 0;

printf("Input String: ");

gets\_s(s);

n = strlen(s);

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

{

//Xet tat ca chuoi con cua s tinh tu vi tri i den j

repeated = 0;

for (j = i + 1; (j < n) && !repeated; j++)

{

for (k = i; (k < j) && !repeated; k++)

{

if (s[j] == s[k])

{

length = j - i;

repeated = 1;

mlem[i] = length;

}

}

}

if (length > longest)longest = length;

}

//mlem mlem

printf("Longest String = %d\n", longest);

while (mlem[t] != '\0')

{

if (mlem[t] == longest)

{

printf("Start Position: %d\n", t);

for (int a = t; a < longest + t; a++)

printf("%c", s[a]);

//printf("\n");

}t++;

}getchar();

}

## Nhập nhiều chuỗi và sắp xếp chúng theo chiều tăng dần

int main() {

char t[20];

int i, j;

int n;

printf("nhap so luong chuoi: ");

scanf("%d", &n);

char s[10][20];

printf("\nNhap chuoi bat ky: \n");

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

scanf("%s", s[i]);

}

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

for (j = 1; j < n; j++) {

if (strcmp(s[j - 1], s[j]) > 0) {

strcpy(t, s[j - 1]);

strcpy(s[j - 1], s[j]);

strcpy(s[j], t);

}

}

}

printf("\nSap xep thu tu tang dan cua cac chuoi:");

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

printf("\n%s", s[i]);

}

return 0;

}

## Tìm từ ngắn nhất và từ dài nhất trong 1 chuỗi dài

void main()

{

char string[] = "Hardships often prepare ordinary people for an extraordinary destiny";

char words[100][100], small[100], large[100];

int i = 0, j = 0, k, length;

for (k = 0; string[k] != '\0'; k++) {

if (string[k] != ' ' && string[k] != '\0') {

words[i][j++] = string[k];

}

else {

words[i][j] = '\0';

i++;

j = 0;

}

}

length = i + 1;

strcpy(small, words[0]);

strcpy(large, words[0]);

for (k = 0; k < length; k++) {

if (strlen(small) > strlen(words[k])) {

strcpy(small, words[k]);

}

if (strlen(large) < strlen(words[k]))

strcpy(large, words[k]);

}

printf("Smallest word: %s\n", small);

printf("Largest word: %s", large);

}

## Ma trận đối xứng

int kiemtradoixung(int a[max][max], int n)

{

int k, j;

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

for (j = 0; j < n && a[k][j] == a[j][k]; j++);

return k == j;

}// ra 1 là đối xứng; ra 0 là ko đối xứng

## Liệt kê tích số nguyên tố từ 2 đến n với n nhập từ bàn phím

void main()

{

int N;

scanf\_s("%d", &N);

for (int i = 2; i <= N; i++) {

int ngt = 2;

int a = i;

int dem = 0;

printf("%d = ", a);

while (a > 1) {

if (a % ngt == 0) {

dem++;

if ((a / ngt) % ngt != 0) {

printf("%d", ngt);

if (dem > 1)printf("^%d ", dem);

if (dem == 1)printf(" ");

if (a / ngt > 1)printf("x ");

}

a /= ngt;

}

else {

ngt++;

dem = 0;

}

}

printf("\n");

}

printf("\n");

}

## Chuyển hoa thành thường và thường thành hoa trong 1 chuỗi

void ChangeCase(char\* s)

{

for (int i = 0; s[i]; i++)

if (islower(s[i])) s[i] = toupper(s[i]);

else s[i] = tolower(s[i]);

}

Ví dụ: Vi Tích Phân -> vI tÍCH pHÂN