**TỔNG HỢP CÁC CODE SOURCE MÔN PRF192**

**MỤC LỤC**

[**I.** **KIẾN THỨC CHUNG** 4](#_Toc140410325)

[1. Kiểu dữ liệu và khai báo: 4](#_Toc140410326)

[2. Toán tử: 4](#_Toc140410327)

[3. Toán so sánh & Logic 4](#_Toc140410328)

[**II.** **CÁC SOURCE CÓ TRONG PE PRF192** 5](#_Toc140410329)

[**Source 1: Đổi vị trí số lẻ lớn nhất và nhỏ nhất** 5](#_Toc140410330)

[**Source 2: Hàm kiểm tra năm nhuận** 7](#_Toc140410331)

[**Source 3: Hàm kiểm tra số Amstrong** 9](#_Toc140410332)

[**Source 4: Vẽ hình vuông** 11](#_Toc140410333)

[**Source 5: Vẽ hình vuông bị rỗng dưới** 12](#_Toc140410334)

[**Source 6: Vẽ hình vuông rỗng trên** 14](#_Toc140410335)

[**Source 7: Vẽ hình vuông rỗng giữa** 16](#_Toc140410336)

[**Source 8: Vẽ hình tam giác** 18](#_Toc140410337)

[**Source 8.1: Vẽ Tam giác ngược 1** 19](#_Toc140410338)

[**Source 9: Vẽ Tam giác ngược 2** 20](#_Toc140410339)

[**Source 10: Vẽ Tam giác ngược 3** 21](#_Toc140410340)

[**Source 11: Vẽ tam giác cân 1** 22](#_Toc140410341)

[**Source 12: Vẽ tam giác cân 2** 23](#_Toc140410342)

[**Source 13.1: Tam giác có số (1)** 24](#_Toc140410343)

[**Source 13.2: Vẽ tam giác có số (2)** 25](#_Toc140410344)

[**Source 13.3: Vẽ tam giác chữ đối xứng** 26](#_Toc140410345)

[**Source 14: Vẽ hình bình hành** 27](#_Toc140410346)

[**Source 15: Vẽ ngôi nhà** 28](#_Toc140410347)

[**Source 16: Vẽ hình thoi** 30](#_Toc140410348)

[**Source 17: Vẽ tam giác rỗng** 32](#_Toc140410349)

[**Source 18: Tính tổng các thàh phần trong mảng** 33](#_Toc140410350)

[**Source 19: Đến số nguyên âm trong mảng** 34](#_Toc140410351)

[**Source 20: Tìm số nhỏ nhất, số lớn nhất** 36](#_Toc140410352)

[**Source 21: Tìm số nhỏ, lớn thứ 2 trong mảng** 38](#_Toc140410353)

[**Source 22: Đảo ngược số** 40](#_Toc140410354)

[**Source 23: Sắp xếp phần tử theo thứ tự tăng dần** 41](#_Toc140410355)

[**Source 24: Sắp xếp phần tử theo thứ tự giảm dần** 42](#_Toc140410356)

[**Source 25: giam bot so luong trung nhau cua mang** 43](#_Toc140410357)

[**Source 26: Vong lap tra ra so lan xuat hien nhieu nhat cua mot phan tu trong arr** 45](#_Toc140410358)

[**Source 27: Tính toán giờ, phút, giây** 48](#_Toc140410359)

[**Source 28: Bình phương các số lẻ trong mảng** 49](#_Toc140410360)

[**Source 29: Hàm kiểm tra các số nguyên tố (Prime Number)** 50](#_Toc140410361)

[**Source 30: Hàm tính tổng n số.** 51](#_Toc140410362)

[**Source 31: Hàm kiểm tra số chính phương** 52](#_Toc140410363)

[**Source 32: Kiểm tra số có phải thuộc dãy Fibonacci** 53](#_Toc140410364)

[**Source 33: Các công thức tính diện tích (S), chu vi (P) hình chữ nhật, tam giác, tròn** 55](#_Toc140410365)

[**Source 34: Sắp xếp theo kiểu Insertion** 56](#_Toc140410366)

[**Source 35: Sắp xếp theo kiểu Bubble** 58](#_Toc140410367)

[**Source 36: Sắp xếp giảm dần (Float)** 60](#_Toc140410368)

[**Source 37: Sắp xếp theo kieu selection** 62](#_Toc140410369)

[**Source 38: Xoa cac phan tu la so le trung nhau** 63](#_Toc140410370)

[**Source 39: ham tinh do dai cua chuoi** 64](#_Toc140410371)

[**Source 40: In ra so chu so trong string** 65](#_Toc140410372)

[**Source 41: In ra so chan lon nhat trong n so** 66](#_Toc140410373)

[**Source 42: Bội chung nhỏ nhất (LCM), Ước chung lớn nhất (GCD)** 67](#_Toc140410374)

[**Source 43: In ra so chan tai vi tri chan va sort** 68](#_Toc140410375)

[**Source 44: Neu la so le, binh phuong roi cong vao** 70](#_Toc140410376)

[**Source 45: In ra tich cua cac so trong chu so n** 71](#_Toc140410377)

[**Source 46: Kiem tra 1 so co phai power of 2** 72](#_Toc140410378)

[**Source 47: In ra ky tu viet hoa at position even** 73](#_Toc140410379)

[**Source 48: In ra cac ky tu la chu** 74](#_Toc140410380)

[**Source 49: Perfect number** 75](#_Toc140410381)

[**Source 50: Dem digit trong string** 76](#_Toc140410382)

[**Source 51: Viet hoa 1/2 length string** 77](#_Toc140410383)

[**Source 52: Check valid name . OUTPUT:1 neu valid** 78](#_Toc140410384)

[**Source 53: Average positive number** 79](#_Toc140410385)

[**Source 54: Diện tích hình tròn** 80](#_Toc140410386)

[**Source 55: check square number** 81](#_Toc140410387)

[**Source 56: xoa 1 phan tu tai vi tri ma nguoi dung nhap vao** 82](#_Toc140410388)

[**Source 57: Chen 1 so bat ky vao array** 84](#_Toc140410389)

[**Source 58: swap 2 vi tri trong array** 86](#_Toc140410390)

[**Source 59: xoa 1 phan tu trong array vi tri nguoi dung nhap** 87](#_Toc140410391)

[**Source 60: In ra gio phut giay** 89](#_Toc140410392)

[**Source 61: Tong so dau va so cuoi** 90](#_Toc140410393)

[**Source 62: Kiem tra exponent** 91](#_Toc140410394)

[**Source 63: Kiem tra n%5==0 in ra so lan xuan hien cua n** 92](#_Toc140410395)

[**Source 64: binary->octal** 93](#_Toc140410396)

[**Source 65: check mang tang dan** 94](#_Toc140410397)

[**Source 66: check mang doi xung** 95](#_Toc140410398)

[**Source 67: In ra số Nguyên tố đầu tiên** 96](#_Toc140410399)

[**Source 68: tính: (-1)n+1\*xn** 97](#_Toc140410400)

[**Source 69: reverse day so** 98](#_Toc140410401)

[**Source 70: square odd number** 99](#_Toc140410402)

[**Source 71: in ra vi tri co tong bang` mot so nao` do** 100](#_Toc140410403)

[**Source 72: Diện tích và chu vi hình chữ nhật** 101](#_Toc140410404)

[**Source 73: Diện tích và chu vi tam giác:** 102](#_Toc140410405)

[**Source 74: Diện tích và chu vi hình tròn:** 103](#_Toc140410406)

[**Source 75: Tính giai thừa F1(n) = n!** 104](#_Toc140410407)

[**Source 76: Tinh tong phan so F2(n) = 1 + 2 + 3 + ... + n** 105](#_Toc140410408)

[**Source 77: Nhap so nguyen n** 106](#_Toc140410409)

[**Source 78: In ra cac so chan o vi tri le theo thu tu giam dan coi v? tri tri dau la vi tri 1** 107](#_Toc140410410)

[**Source 79: Nhập chuỗi và in ra từ dài nhất và ngắn nhất** 109](#_Toc140410411)

1. **KIẾN THỨC CHUNG**
2. Kiểu dữ liệu và khai báo:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu dữ liệu** | **Khai báo** | **Kích thước** |
| 1 | int | %d, %i | 2 hoặc 4 bytes |
| 2 | sa | %c | 1 bytes |
| 3 | float | %f, %.(n)f (làm tròn đến chữ số thứ n sau dấu thập phân) | 4 bytes |
| 4 | double | %lf | 8 bytes |
| 5 | short int | %hd | 2 bytes |
| 6 | unsigned int | %u | 2 hoặc 4 bytes |
| 7 | long int | %ld, %li | 4 bytes |
| 8 | long long int | %lld, %lli |  |
| 9 | unsigned long int | %lu | 4 bytes |
| 10 | unsigned long long int | %llu | 8 bytes |
| 11 | signed char | %c | 1 bytes |
| 12 | unsigned char | %c | 1 bytes |
| 13 | long double | %lf | 10 bytes |

1. Toán tử:

|  |  |  |
| --- | --- | --- |
| **Phép toán** | **Ý nghĩa** | **Ví dụ** |
| + | Cộng | a+b |
| - | Trừ | a-b |
| \* | Nhân | a\*b |
| / | chia | a/b |
| % | Chia lấy phần dư | a%b |

1. Toán so sánh & Logic

|  |  |
| --- | --- |
| **Phép toán** | **Ý nghĩa** |
| > | Lớn hơn |
| < | Nhỏ hơn |
| >= | Lớn hoặc bằng |
| <= | Nhỏ hoặc bằng |
| == | Bằng |
| != | khác |

1. **CÁC SOURCE CÓ TRONG PE PRF192**

## **Source 1: Đổi vị trí số lẻ lớn nhất và nhỏ nhất**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <ctype.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

system("cls");

//INPUT - @STUDENT:ADD YOUR CODE FOR INPUT HERE:

int n;

scanf("%d", &n);

int\* arr = (int\*)malloc(n \* sizeof(int)); // Dynamically allocate memory

int i;

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

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

}

int min\_odd = -1, max\_odd = -1;

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

if (arr[i] % 2 != 0) {

if (min\_odd == -1 || arr[i] < arr[min\_odd]) {

min\_odd = i;

}

if (max\_odd == -1 || arr[i] > arr[max\_odd]) {

max\_odd = i;

}

}

}

if (min\_odd != -1 && max\_odd != -1) {

swap(&arr[min\_odd], &arr[max\_odd]);

}

// Fixed Do not edit anything here.

printf("\nOUTPUT:\n");

//@STUDENT: WRITE YOUR OUTPUT HERE:

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

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

}

printf("\n");

free(arr);

//--FIXED PART - DO NOT EDIT ANY THINGS HERE

printf("\n");

system ("pause");

return(0);

}

TEST CASE:

A black screen with white text

Description automatically generated

## **Source 2: Hàm kiểm tra năm nhuận**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <ctype.h>

int isLeapYear(int year) {

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0)

return 1;

else

return 0;

}

else

return 1;

}

else

return 0;

}

int main() {

system("cls");

//INPUT - @STUDENT:ADD YOUR CODE FOR INPUT HERE:

int year;

scanf("%d", &year);

// Fixed Do not edit anything here.

printf("\nOUTPUT:\n");

//@STUDENT: WRITE YOUR OUTPUT HERE:

if (isLeapYear(year))

printf("%d is a leap year.", year);

else

printf("%d is not a leap year.", year);

//--FIXED PART - DO NOT EDIT ANY THINGS HERE

printf("\n");

system ("pause");

return(0);

}

TEST CASE:

A black background with white text

Description automatically generatedA black background with white text

Description automatically generated

## **Source 3: Hàm kiểm tra số Amstrong**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <stdbool.h>

int countDigits(int n) {

int count = 0;

while (n > 0){

n /= 10;

count++;

}

return count;

}

bool isArmstrong(int n) {

int numDigits = countDigits(n);

int tmp = n;

int sum = 0;

int last;

while (tmp > 0) {

last = tmp % 10;

tmp /= 10;

sum += pow(last, numDigits);

}

return sum == n;

}

int main() {

system("cls");

int n;

scanf("%d", &n);

printf("\nOUTPUT:\n");

if (isArmstrong(n)) {

printf("%d is an Armstrong number.", n);

} else {

printf("%d is not an Armstrong number.", n);

}

printf("\n");

system("pause");

return 0;

}

TEST CASE:

A black background with white text

Description automatically generatedA black background with white text

Description automatically generated

## **Source 4: Vẽ hình vuông**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void veHinhVuong1(int n) {

int i, j;

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

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

printf("\* ");

}

printf("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veHinhVuong1(n);

return 0;

}

TEST CASE:

A screen shot of a black background

Description automatically generated

## **Source 5: Vẽ hình vuông bị rỗng dưới**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void veHinhVuong(int n){

int i, j;

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

if (i == 0) {

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

printf("\* ");

}

} else {

printf("\*");

for (j = 0; j < n - 2; j++) {

printf(" ");

}

printf(" \*");

}

printf("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veHinhVuong(n);

return 0;

}

TEST CASE:

A screen shot of a black screen

Description automatically generated

## **Source 6: Vẽ hình vuông rỗng trên**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void veHinhVuong(int n){

int i, j;

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

if (i == n-1) {

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

printf("\* ");

}

} else {

printf("\*");

for (j = 0; j < n - 2; j++) {

printf(" ");

}

printf(" \*");

}

printf("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veHinhVuong(n);

return 0;

}

TEST CASE:

A black screen with white text

Description automatically generated

## **Source 7: Vẽ hình vuông rỗng giữa**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void veHinhVuong (int n){

int i,j;

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

if (i==n-1 || i==0 ) {

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

printf ("\* ");

}

}

else {

printf ("\*");

for (j=0;j<2\*n-3;j++){

printf (" ");

}

printf ("\*");

}

printf ("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veHinhVuong(n);

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 8: Vẽ hình tam giác**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void veTamGiac (int n){

int i,j;

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

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

printf ("\* ");

}

printf ("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veTamGiac(n);

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 8.1: Vẽ Tam giác ngược 1**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void veTamGiac (int n){

int i,j;

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

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

printf ("\* ");

}

printf ("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veTamGiac(n);

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 9: Vẽ Tam giác ngược 2**

#include <stdio.h>

void veTamGiac(int n) {

int i, j, size = n;

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

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

printf(" ");

}

for (j = i; j < size; j++) {

printf("\*");

}

printf("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veTamGiac(n);

return 0;

}

TEST CASE:

A black background with white stars

Description automatically generated

## **Source 10: Vẽ Tam giác ngược 3**

#include <stdio.h>

void veTamGiac (int n){

int i,j;

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

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

printf (" ");

}

for (j=0;j<i+1;j++){

printf ("\*");

}

printf ("\n");

}

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

veTamGiac(n);

return 0;

}

TEST CASE:

A black background with white stars

Description automatically generated

## **Source 11: Vẽ tam giác cân 1**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

int i, space, rows, k = 0;

scanf("%d", &rows);

for (i = 1; i <= rows; ++i, k = 0) {

for (space = 1; space <= rows - i; ++space) {

printf(" ");

}

while (k != 2 \* i - 1) {

printf("\* ");

++k;

}

printf("\n");

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 12: Vẽ tam giác cân 2**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

int rows, i, j, space;

scanf("%d", &rows);

for (i = rows; i >= 1; --i) {

for (space = 0; space < rows - i; ++space)

printf(" ");

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

printf("\* ");

for (j = 0; j < i - 1; ++j)

printf("\* ");

printf("\n");

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 13.1: Tam giác có số (1)**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

int rows, i, j, number = 1;

scanf("%d", &rows);

printf("OUTPUT:\n");

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

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

printf("%d ", number);

++number;

}

printf("\n");

}

return 0;

}

TEST CASE:

A black screen with white text

Description automatically generated

## **Source 13.2: Vẽ tam giác có số (2)**

#include <stdio.h>

int main(){

int i, j, row;

scanf("%d", &row);

printf("OUTPUT:\n");

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

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

printf("%d ", j);

}

printf("\n");

}

return 0;

}

TEST CASE:

**A black background with white text

Description automatically generated**

## **Source 13.3: Vẽ tam giác chữ đối xứng**

#include <stdio.h>

int main() {

int i, j, rows, count=0;

scanf("%d", &rows);

printf("OUTPUT:\n");

for (i = 0; i < 2\*rows; i=i+2) {

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

printf("%c", 'A'+count);

if(j < i/2)

count++;

else

count--;

}

count = 0;

printf("\n");

}

return(0);

}

TEST CASE:

A black screen with white text

Description automatically generated

## **Source 14: Vẽ hình bình hành**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

int i, j, n;

scanf("%d", &n);

printf("OUTPUT:\n");

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

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

printf(" ");

}

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

if (i == 1 || j == 1 || i == n || j == n) {

printf("\*");

} else {

printf(" ");

}

}

printf("\n");

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 15: Vẽ ngôi nhà**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

int i, j, space, rows = 8, star = 0;

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

{

if (i < 5)

{

for (space = 1; space < 5 - i; space++)

{

printf(" ");

}

while (star != (2 \* i + 1))

{

printf("\*");

star++;

;

}

star = 0;

/\* move to next row \*/

printf("\n");

}

else

{

for (j = 0; j < 9; j++)

{

if ((int)(j / 3) == 1)

printf(" ");

else

printf("\*");

}

printf("\n");

}

}

return 0;

}

TEST CASE

A black background with white dots

Description automatically generated

## **Source 16: Vẽ hình thoi**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

int i;

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

{

int j;

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

{

printf(" ");

}

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

{

printf(" \* ");

}

printf("\n");

}

for (i = n - 1; i >= 1; i--)

{

int j;

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

{

printf(" ");

}

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

{

printf(" \* ");

}

printf("\n");

}

return 0;

}

TEST CASE:

A screen shot of a computer screen

Description automatically generated

## **Source 17: Vẽ tam giác rỗng**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

int i, j, n;

scanf("%d", &n);

printf("OUTPUT:\n");

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

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

if (j == 1 || i == n || i == j)

printf("\*");

else

printf(" ");

}

printf("\n");

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 18: Tính tổng các thàh phần trong mảng**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <stdbool.h>

int main() {

system("cls");

int n;

scanf("%d", &n);

int a[n];

int i;

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

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

}

int sum=0;

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

sum+=a[i];

}

printf("\nOUTPUT:\n");

printf("%d", sum);

printf("\n");

system("pause");

return 0;

}

TEST CASE:

A black screen with white text

Description automatically generated

## **Source 19: Đến số nguyên âm trong mảng**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <stdbool.h>

int main() {

system("cls");

int n;

scanf("%d", &n);

int a[n];

int i;

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

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

}

int count=0;

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

if (a[i]<0) {

count++;

}

}

printf("\nOUTPUT:\n");

printf("%d", count);

printf("\n");

system("pause");

return 0;

}

TEST CASE:

A black screen with white text

Description automatically generated

## **Source 20: Tìm số nhỏ nhất, số lớn nhất**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <stdbool.h>

int main() {

system("cls");

int n;

scanf("%d",&n);

int i,Arr[n];

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

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

int min = Arr[0];

int max = Arr[0];

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

if (Arr[i] < min) {

min = Arr[i];

}

}

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

if (Arr[i] > max) {

max = Arr[i];

}

}

printf("\nOUTPUT:\n");

printf ("%d %d",min, max);

printf("\n");

system("pause");

return 0;

}

TEST CASE:

A black screen with white text

Description automatically generated

## **Source 21: Tìm số nhỏ, lớn thứ 2 trong mảng**

#include <stdio.h>

#include <stdlib.h>

void findSecondLargestAndSmallest(int arr[], int size) {

int max = arr[0];

int secondMax = arr[0];

int min = arr[0];

int secondMin = arr[0];

int i;

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

if (arr[i] > max) {

secondMax = max;

max = arr[i];

} else if (arr[i] > secondMax && arr[i] != max) {

secondMax = arr[i];

}

if (arr[i] < min) {

secondMin = min;

min = arr[i];

} else if (arr[i] < secondMin && arr[i] != min) {

secondMin = arr[i];

}

}

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

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

}

int main() {

int n;

scanf("%d", &n);

int\* arr = (int\*)malloc(n \* sizeof(int));

int i;

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

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

}

printf("OUTPUT:\n");

findSecondLargestAndSmallest(arr, n);

free(arr);

return 0;

}TEST CASE:

A black screen with white text

Description automatically generated

## **Source 22: Đảo ngược số**

#include <stdio.h>

#include <stdlib.h>

int main() {

int i,j;

long n;

do

scanf("%d",&n);

while (n<0);

long rev=0;

while (n!=0){

rev=rev\*10+n%10;

n/=10;

}

n=rev;

printf("OUTPUT:\n");

printf("%d",n);

return 0;

}

TEST CASE:

A black background with white numbers

Description automatically generated

## **Source 23: Sắp xếp phần tử theo thứ tự tăng dần**

#include <stdio.h>

#include <stdlib.h>

int main() {

int n;

scanf("%d", &n);

int a[n];

int i;

for(i=0; i<n; i++) scanf("%d", &a[i]);

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

int j;

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

if (a[j] > a[j+1]) {

int temp = a[j];

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

a[j+1] = temp;

}

}

}

printf("OUTPUT:\n");

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

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

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 24: Sắp xếp phần tử theo thứ tự giảm dần**

#include <stdio.h>

#include <stdlib.h>

int main() {

int n;

scanf("%d", &n);

int a[n];

int i;

for(i=0; i<n; i++) scanf("%d", &a[i]);

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

int j;

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

if (a[j] < a[j+1]) {

int temp = a[j];

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

a[j+1] = temp;

}

}

}

printf("OUTPUT:\n");

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

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

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 25: giam bot so luong trung nhau cua mang**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <ctype.h>

int main() {

system("cls");

//INPUT - @STUDENT:ADD YOUR CODE FOR INPUT HERE:

int n;

int arr[1000], b[1000] = {0}, c[1000] = {0}; //Khoi tao 3 mang so nguyen

int i;

scanf("%d", &n); //Nhap so phan tu cua mang arr

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

{

scanf("%d", &arr[i]); // Nhap n phan tu mang so nguyen arr

}

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

{

if(arr[i] > 0) //Mang so nguyen b[] ghi lai so lan xuat hien cua

b[arr[i]]++; //cac phan tu lon hon 0 trong day arr

if(arr[i] < 0) //Mang so nguyen c[] ghi lai so lan xuat hien cua

c[-arr[i]]++; //cac phan tu nho hon 0 trong day arr

}

int max = 0;

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

{

if(arr[i] > 0)

{

if(b[arr[i]] > max)

max = b[arr[i]];

}

else

{

if(c[-arr[i]] > max)

max = c[-arr[i]];

}

}

## **Source 26: Vong lap tra ra so lan xuat hien nhieu nhat cua mot phan tu trong arr**

// Fixed Do not edit anything here.

printf("\nOUTPUT:\n");

//@STUDENT: WRITE YOUR OUTPUT HERE:

for(i = 0; i <1000 ;i++)

{

if(b[i] == max)

{

printf("%d ", i);

}

if(c[i] == max)

printf("%d ", -i);

} //In ra phan tu co lan xuat hien nhieu nhat trong day

//--FIXED PART - DO NOT EDIT ANY THINGS HERE

printf("\n");

system ("pause");

return(0);

}

\

-------------------------------------------------------------------------------------------------------------------------------------------

// kiem tra can bac 2 cua 1 so

int a, tmp, flag = 0;

scanf("%d", &a);

tmp = a;

while (a % 2 == 0)

{

a /= 2;

flag++;

}

printf("OUTPUT:\n");

if (a == 1)

{

printf("%d", flag);

}

else

{

printf("%d is not a power of 2", tmp);

}

//dem so luong tu cua 3 tu dau tien

char s[100];

int i, count = 0, tmp = 0;

gets(s);

printf("OUTPUT:\n");

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

{

if (s[i] != ' ' && tmp < 3)

{

while (s[i] != ' ')

{

count++;

++i;

;

}

tmp++;

}

}

printf("%d", count);

## **Source 27: Tính toán giờ, phút, giây**

#include<stdio.h>

int main(){

int a;

scanf("%d",&a);

printf("OUTPUT:\n");

int h,m,s;

h=a/3600;

m=(a-(h\*3600))/60;

s=a-h\*3600-m\*60;

printf("%d:%d:%d", h, m, s);

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 28: Bình phương các số lẻ trong mảng**

#include <stdio.h>

int main() {

int size;

scanf("%d",&size);

int i,Arr[size];

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

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

printf("OUTPUT:\n");

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

if(Arr[i]%2==1&& Arr[i]>0) {

printf("%d ",Arr[i]\*Arr[i]);

}

return 0;

}TEST CASE:

A black background with white text

Description automatically generated

## **Source 29: Hàm kiểm tra các số nguyên tố (Prime Number)**

#include <stdio.h>

#include <math.h>

int Prime(int n) {

int i;

if (n < 2) return 0;

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

if (n % i == 0) return 0;

}

return 1;

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

if (Prime(n)) {

printf("%d is a prime number.\n", n);

} else {

printf("%d is not a prime number.\n", n);

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated A black background with white text

Description automatically generated

## **Source 30: Hàm tính tổng n số.**

#include <stdio.h>

#include <math.h>

int SumInRange(int n) {

int i;

int sum = 0;

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

sum += i;

}

return sum;

}

int main() {

int n;

int sum;

scanf("%d", &n);

printf("OUTPUT:\n");

sum = SumInRange(n);

printf("%d", sum);

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 31: Hàm kiểm tra số chính phương**

#include <stdio.h>

#include <math.h>

int SCP(int n) {

if (sqrt(n) == (int)sqrt(n))

return 1;

else

return 0;

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

if (SCP(n))

printf("%d is a perfect square.\n", n);

else

printf("%d is not a perfect square.\n", n);

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated A black background with white text

Description automatically generated

## **Source 32: Kiểm tra số có phải thuộc dãy Fibonacci**

#include <stdio.h>

#include <math.h>

int isPerfectSquare(int n) {

int sqrtN = sqrt(n);

return (sqrtN \* sqrtN == n);

}

int isFibonacci(int n) {

return isPerfectSquare(5 \* n \* n + 4) || isPerfectSquare(5 \* n \* n - 4);

}

int main() {

int n;

scanf("%d", &n);

printf("OUTPUT:\n");

if (isFibonacci(n)) {

printf("%d is a Fibonacci number.\n", n);

} else {

printf("%d is not a Fibonacci number.\n", n);

}

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated A black background with white text

Description automatically generated

## **Source 33: Các công thức tính diện tích (S), chu vi (P) hình chữ nhật, tam giác, tròn**

#include <stdio.h>

#include <math.h>

double getPerimCircle(double r) {

if (r > 0)

return 2 \* r \* 3.14159; // S? d?ng giá tr? pi chính xác hon

else

return 0;

}

double getAreaCircle(double r) {

if (r > 0)

return r \* r \* 3.14159;

else

return 0;

}

double getPerimRectangle(double A, double B) {

if (A > 0 && B > 0)

return 2 \* (A + B);

else

return 0;

}

double getAreaRectangle(double A, double B) {

if (A > 0 && B > 0)

return A \* B;

else

return 0;

}

double getPerimTriangle(double A, double B, double C) {

if (A > 0 && B > 0 && C > 0 && A + B > C && B + C > A && A + C > B)

return A + B + C;

else

return 0;

}

double getAreaTriangle(double A, double B, double C) {

if (A > 0 && B > 0 && C > 0 && A + B > C && B + C > A && A + C > B) {

double P = getPerimTriangle(A, B, C) / 2;

return sqrt(P \* (P - A) \* (P - B) \* (P - C));

} else {

printf("\nÐây không ph?i là tam giác, vui lòng ki?m tra d? dài các c?nh A, B, C\n");

return 0;

}

}

int main() {

double r = 5.0; // Bán kính c?a hình tròn

double A = 3.0, B = 4.0, C = 5.0; // Các c?nh c?a tam giác

printf("Chu vi c?a Hình Tròn: %lf\n", getPerimCircle(r));

printf("Di?n tích c?a Hình Tròn: %lf\n", getAreaCircle(r));

printf("Chu vi c?a Hình Ch? Nh?t: %lf\n", getPerimRectangle(A, B));

printf("Di?n tích c?a Hình Ch? Nh?t: %lf\n", getAreaRectangle(A, B));

printf("Chu vi c?a Tam Giác: %lf\n", getPerimTriangle(A, B, C));

printf("Di?n tích c?a Tam Giác: %lf\n", getAreaTriangle(A, B, C));

return 0;

}

## **Source 34: Sắp xếp theo kiểu Insertion**

#include <stdio.h>

void Insertion(int A[], int n) {

int i, j;

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

int value = A[i];

j = i - 1;

while (j >= 0 && value < A[j]) {

A[j + 1] = A[j];

j--;

}

A[j + 1] = value;

}

}

int main() {

int n;

scanf("%d", &n);

int A[n];

int i;

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

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

}

Insertion(A, n);

printf("OUTPUT:\n");

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

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

}

printf("\n");

return 0;

}

TEST CASE:

A black background with white numbers

Description automatically generated

## **Source 35: Sắp xếp theo kiểu Bubble**

#include <stdio.h>

void bubble(int A[], int n, int switchOper) {

int i, j;

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

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

if ((A[j] - A[j + 1]) \* switchOper > 0) {

int temp = A[j];

A[j] = A[j + 1];

A[j + 1] = temp;

}

}

}

}

int main() {

int n;

scanf("%d", &n);

int A[n];

int i;

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

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

}

bubble(A, n, 1);

printf("OUTPUT:\n");

printf("Sắp xếp tang dan: ");

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

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

}

printf("\n");

bubble(A, n, -1);

printf("Sắp xếp giam dan: ");

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

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

}

printf("\n");

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 36: Sắp xếp giảm dần (Float)**

#include <stdio.h>

#include <stdlib.h>

void bubble(float A[], int n, int switchOper) {

int i, j;

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

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

if ((A[j] - A[j + 1]) \* switchOper > 0) {

float temp = A[j];

A[j] = A[j + 1];

A[j + 1] = temp;

}

}

}

}

int main() {

int n;

scanf("%d", &n);

float Arr[n];

int i;

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

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

}

printf("OUTPUT:\n");

bubble(Arr, n, -1);

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

printf("%.2f ", Arr[i]);

}

return 0;

}TEST CASE:

A black background with white numbers

Description automatically generated

## **Source 37: Sắp xếp theo kieu selection**

void selection(int A[],int size,int opt){

int i,j;

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

int min\_pos=i;

for(j=i+1;j<size;j++)

if((A[min\_pos]-A[j])\*opt>0)

min\_pos=j;

if(i!=min\_pos){

int temp=A[i];

A[i]=A[min\_pos];

A[min\_pos]=temp;

}

}

}

## **Source 38: Xoa cac phan tu la so le trung nhau**

void DelEle(int Arr[],int size,int Pos){

int i;

for(i=Pos;i<size-1;i++)

Arr[i]=Arr[i+1];

size--;

}

int main(){

int i,j,size;

do {

scanf("%d", &size);

}while ( size>=10);

int Arr[size];

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

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

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

for(j=i+1;j<size;j++)

if(Arr[i]==Arr[j] && Arr[j]%2!=0)

{

DelEle(Arr,size,j);

size--;

j--; // descrease size and iterator

}

printf("\nOUTPUT:\n");

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

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

return 0;

}

## **Source 39: ham tinh do dai cua chuoi**

int string\_length(char s[]) {

int i=0;

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

i++;

return i;

}

// ham dao nguoc chuoi

void string\_reverse(char str[]) {

int i, j, len;

char ch;

j = len = string\_length(str) - 1;

i = 0;

while(i < j) {

ch = str[j];

str[j] = str[i];

str[i] = ch;

i++;

j--;

}

}

## **Source 40: In ra so chu so trong string**

#include <stdio.h>

#include<string.h>

int main (){

char str[100];

printf("Please enter a string:");

gets(str);

int i,count=0;

int len;

len=strlen(str);

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

{

if(str[i]>48 && str[i]<57)

count++;

}

printf("%d",count);

return 0;

}

## **Source 41: In ra so chan lon nhat trong n so**

#include <stdio.h>

#include<string.h>

int main (){

// int size;

// scanf("%d",&size);

int i,Arr[5];

for(i=0;i<5;i++)

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

int Max\_even=-1;

for(i=0;i<5;i++)

if(Arr[i]>Max\_even && Arr[i]%2==0)

Max\_even=Arr[i];

//printf("OUTPUT:\n")

printf("\nOUTPUT:\n%d",Max\_even);

return 0;

}

## **Source 42: Bội chung nhỏ nhất (LCM), Ước chung lớn nhất (GCD)**

#include<stdio.h>

int GCD(int a, int b){

while (a!=b)

if(a>b) a=a-b;

else b=b-a;

return b;

}

int LCM(int a, int b){

int lcm=a \* b /GCD(a,b);

return lcm;

}

int main(){

int a,b;

printf("Please enter two positive integer to find GCD,LCM: ");

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

GCD(a,b);

printf("OUTPUT:\n");

printf("%d\t%d", GCD(a,b),LCM(a,b));

return 0;

}

## **Source 43: In ra so chan tai vi tri chan va sort**

#include <stdio.h>

void printsx(int A[], int\* n)

{

int \*adds[\*n];

int i,j;

int\* temp;

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

{

adds[i]=&A[i];

}

for(j=0;j<\*n-1;j++)

{

if(adds[j]>adds[j+1])

{

temp=adds[j];

adds[j]=adds[j+1];

adds[j+1]=temp;

}

}

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

{

if(\*adds[i]%2==0 && i%2==0 && i!=0)

printf("%d\t", \*adds[i]);

}

}

int main(){

int i,size;

scanf("%d",&size);

int Arr[size];

Arr[0]=size;

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

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

printsx(Arr,&size);

return 0;

}

## **Source 44: Neu la so le, binh phuong roi cong vao**

#include <stdio.h>

int main(){

int size;

int sum=0;

do {

scanf("%d",&size);

}while(size>10);

int i,Arr[size];

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

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

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

if(Arr[i]%2!=0)

sum=sum+(Arr[i]\*Arr[i]);

printf("%d",sum);

return 0;

}

## **Source 45: In ra tich cua cac so trong chu so n**

#include <stdio.h>

int main(){

int n,a,pro=1;

scanf("%d",&a);

do {

n=a%10;

pro\*=n;

a=a/10;

}while (a>0);

printf("%d",pro);

return 0;

}

## **Source 46: Kiem tra 1 so co phai power of 2**

#include<stdio.h>

int main(){

int remainder,a,n=0,a1;

scanf("%d", &a);

a1=a;

printf("\nOUTPUT:\n");

while (a !=0 ){

remainder=a%2;

if(remainder!=0) break;

a/=2;

n++;

}

if(a>1) printf("%d is not power of 2",a1);

else printf("%d", n);

return (0);

}

## **Source 47: In ra ky tu viet hoa at position even**

#include<stdio.h>

#include<string.h>

int main(){

char str[100];

scanf("%s",&str);

int i,len=strlen(str);

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

if(i%2!=0) str[i]-=32;

printf("%s",str);

return (0);

}

## **Source 48: In ra cac ky tu la chu**

#include<stdio.h>

#include<string.h>

int main(){

char str[100];

scanf("%[^\n]",&str);

int i,len;

len=strlen(str);

printf("\nOUTPUT:\n");

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

{

if(str[i]>='a' && str[i]<='z' || str[i]>='A' && str[i]<='Z')

printf("%c",str[i]);

}

return (0);

}

## **Source 49: Perfect number**

#include<stdio.h>

#include<string.h>

int main(){

int n;

scanf("%d",&n);

int i,sum=0;

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

if(n%i==0) sum+=i;

} if (sum==n)

printf("perfect");

else printf("NOT");

return (0);

}

## **Source 50: Dem digit trong string**

#include<stdio.h>

#include<string.h>

int main(){

char str[100];

scanf("%s",&str);

int i;

int len=strlen(str);

int count=0;

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

if(isdigit(str[i])) count++;

} printf("%d",count);

return (0);

}

## **Source 51: Viet hoa 1/2 length string**

#include <stdio.h>

#include<string.h>

int main(){

char str[200];

printf("Enter a string: ");

scanf("%[^\n]",&str);

int i,n=strlen(str)/2;

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

if(str[i]>='A'&&str[i]<='Z') str[i]+=32;

for(i=n;i<strlen(str);i++)

if(str[i]>='a' && str[i]<='z')

str[i]-=32;

}

puts(str);

return 0;

}

## **Source 52: Check valid name . OUTPUT:1 neu valid**

#include<stdio.h>

#include<string.h>

int main(){

char str[100];

fgets(str, sizeof(str), stdin);

int i;

// check extra space

for(i = 0 ; i < strlen(str); i++){

if(str[i] == ' ' && str[i+1] == ' ') {

printf("0");

return 0;

}

}

// check alpha

int count = 0;

// dem nhung thang nao ma khong phai alpha , nhung ki tu dac biet

for(i = 0; i < strlen(str) ; i++){

if(str[i] >= 'a' && str[i] <= 'z' || str[i] >= 'A' && str[i] <= 'Z' || str[i] == ' ' ) {

}

else{

count++;

}

}

if(count-1!=0) printf("0");

else printf("1");

}

## **Source 53: Average positive number( Số dương trung bình)**

#include<stdio.h>

#include<string.h>

double averageNumber(int Arr[], int size){

int i,count=0;

int sum=0;

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

if(Arr[i]>=0) {

sum+=Arr[i];

count++;

}

if(count==0) return 0;

else return sum\*1.0 /count ;

}

int main(){

int size;

scanf("%d",&size);

int i,Arr[size];

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

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

}

printf("%.2lf",averageNumber(Arr,size));

}

## **Source 54: Diện tích hình tròn**

#include<stdio.h>

#define PI 3.14159

int main(){

double a;

printf("Enter radius to calculate the area of a circle:");

scanf("%lf",&a);

double area=0;

if (a>0) area=PI\*a\*a;

printf("%lf",area);

return 0;

}

## **Source 55: check square number**

#include<stdio.h>

#include<math.h>

int SCP(int n){

if (sqrt(n)==(int)sqrt(n)) return 1;

else return 0;

}

int main(){

int a;

scanf("%d",&a);

if(SCP(a)) printf("Pefect number");

else printf("Not");

return 0;

}

## **Source 56: xoa 1 phan tu tai vi tri ma nguoi dung nhap vao**

#include <stdio.h>

#include<time.h>

#include<stdlib.h>

int main(){

int i,size;

srand(time(NULL));

printf("Enter the size of the array: "); scanf("%d", &size);

int A[size];

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

{

A[i]=rand()%51-rand()%30;

}

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

printf("%d\t", A[i]);

int pos;

printf("\nEnter the position to remove an element: "); scanf("%d", &pos);

if(pos>=0 && pos<size) {

int value=A[pos];

for(i=pos;i<size;i++)

A[i]=A[i+1];

size--;

printf("\nThe array after removing element %d at position %d:\n",value,pos);

for(i=0;i<size;i++) printf("%d\t", A[i]);

} else {

printf("\nCannot remove the element at %d, plz check range !", pos);

printf("\nThe origin array:\n");

for(i=0;i<size;i++) printf("%d\t", A[i]);

}

return 0;

}

## **Source 57: Chen 1 so bat ky vao array**

#include <stdio.h>

#include<time.h>

#include<stdlib.h>

int main(){

int i,size;

srand(time(NULL));

printf("Enter the size of the array: "); scanf("%d", &size);

int A[size];

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

{

A[i]=rand()%51-rand()%30;

}

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

printf("%d\t", A[i]);

int pos;

printf("\nEnter the position to insert an element: "); scanf("%d", &pos);

if(pos>=0 && pos<=size) {

for(i=size;i>=pos;i--)

A[i]=A[i-1];

int n;

printf("Enter new element: "); scanf("%d", &n);

A[pos]=n;

printf("\nThe array after insert element %d at position %d:\n",n,pos);

size++;

for(i=0;i<size;i++) printf("%d\t", A[i]);

} else {

printf("\nCannot insert the element at %d, plz check range !", pos);

printf("\nThe origin array:\n");

for(i=0;i<size;i++) printf("%d\t", A[i]);

}

return 0;

}

## **Source 58: swap 2 vi tri trong array**

#include <stdio.h>

#include<stdlib.h>

int main(){

int i,size;

printf("Enter the size of the array: "); scanf("%d", &size);

int A[size];

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

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

int pos1,pos2;

printf("\nEnter the two positions to swap : "); scanf("%d%d", &pos1, &pos2);

if (pos1>=0 && pos2>=0 && pos1<size && pos2<size && pos1 != pos2) {

{

A[pos1]=A[pos1]+A[pos2];

A[pos2]=A[pos1]-A[pos2];

A[pos1]=A[pos1]-A[pos2];

}

printf("\nThe array after swapping :\n",pos1,pos2);

for(i=0;i<size;i++) printf("%d\t", A[i]);

} else {

printf("\nCannot swap the element at position %d and %d, plz check range !", pos1, pos2);

printf("\nThe origin array:\n");

for(i=0;i<size;i++) printf("%d\t", A[i]);

}

return 0;

}

## **Source 59: xoa 1 phan tu trong array vi tri nguoi dung nhap**

#include <stdio.h>

#include<stdlib.h>

int main(){

int i,size;

printf("Enter the size of the array: "); scanf("%d", &size);

int A[size];

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

{

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

}

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

printf("%d\t", A[i]);

int pos;

printf("\nEnter the position to remove an element: "); scanf("%d", &pos);

if(pos>=0 && pos<size) {

int value=A[pos];

for(i=pos;i<size;i++)

A[i]=A[i+1];

size--;

printf("\nThe array after removing element %d at position %d:\n",value,pos);

for(i=0;i<size;i++) printf("%d\t", A[i]);

} else {

printf("\nCannot remove the element at %d, plz check range !", pos);

printf("\nThe origin array:\n");

for(i=0;i<size;i++) printf("%d\t", A[i]);

}

return 0;

}

## **Source 60: In ra gio phut giay**

#include<stdio.h>

int main(){

int a;

scanf("%d",&a);

int h,m,s;

h=a/3600;

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

m=(a-(h\*3600))/60;

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

s=a-h\*3600-m\*60;

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

return 0;

}

## **Source 61: Tong so dau va so cuoi**

#include <stdio.h>

int sum, a[100], x, count1 = 0, i;

int main()

{

scanf("%d", &x);

while (x != 0)

{

int n = x % 10;

x = x / 10;

a[count1++] = n;

}

sum = a[0] + a[count1 - 1];

printf("%d", sum);

return 0;

}

## **Source 62: Kiem tra exponent**

#include <stdio.h>

#include <math.h>

int main() {

int num, base;

double exponent;

printf("Nh?p s? nguyên: ");

scanf("%d", &num);

for (base = 2; base <= num; base++) {

exponent = log(num) / log(base);

if (exponent == (int) exponent) {

printf("%d là l?y th?a c?a %d^%d\n", num, base, (int) exponent);

return 0;

}

}

printf("%d không ph?i là l?y th?a c?a b?t k? s? nào\n", num);

return 0;

}

## **Source 63: Kiem tra n%5==0 in ra so lan xuan hien cua n**

#include<stdio.h>

#define max 100

int main()

{

int i,n;

scanf("%d",&n);

int Arr[n];

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

{

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

}

int count[max] = {0};

printf("\nOUTPUT:\n");

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

{

if(Arr[i]%5==0)

count[Arr[i]]++;

}

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

{

if (count[i] > 0)

{

printf("%d-%d\n", count[i], i);

}

}

return 0;

}

## **Source 64: binary->octal**

#include <stdio.h>

#include <math.h>

int binaryToOctal(long long binaryNumber);

int main() {

long long binaryNumber;

printf("Enter a binary number: ");

scanf("%lld", &binaryNumber);

printf("Octal number = %d", binaryToOctal(binaryNumber));

return 0;

}

int binaryToOctal(long long binaryNumber) {

int octalNumber = 0, decimalNumber = 0, i = 0;

while (binaryNumber != 0) {

decimalNumber += (binaryNumber % 10) \* pow(2, i);

++i;

binaryNumber /= 10;

}

i = 1;

while (decimalNumber != 0) {

octalNumber += (decimalNumber % 8) \* i;

decimalNumber /= 8;

i \*= 10;

}

return octalNumber;

}

## **Source 65: check mang tang dan**

#pragma warning (disable : 4996)

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#define maxn 1000

int check(int a[], int n)

{

int i;

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

{

if (a[i] > a[i + 1]) return 0;

}

return 1;

}

int main()

{

int n, i, j, a[maxn];

scanf("%d", &n);

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

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

if (check(a, n) == 0) printf("-1");

else printf("Mang tang dan");

return 0;

}

## **Source 66: check mang doi xung**

#pragma warning (disable : 4996)

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#include <ctype.h>

#define MAXN 1000

int main()

{

int n, i, a[MAXN], check;

scanf("%d", &n);

for (i = 0; i < n; ++i) scanf("%d", a + i);

for (i = 0; i <= n / 2; ++i)

{

if (a[i] != a[n - i - 1])

{

check = 0;

break;

}

else check = 1;

}

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

else printf("0");

return 0;

}

## **Source 67: In ra số Nguyên tố đầu tiên**

int checkprime(int n)

{

int i;

if (n < 2) return 0;

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

if (n % i == 0) return 0;

return 1;

}

int main()

{

int n, i, j, a[n];

scanf("%d", &n);

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

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

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

if (checkprime(a[i]) == 1)

{

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

break;

}

if (i == n) printf("-1");

return 0;

}

## **Source 68: tính: (-1)n+1\*xn**

#include<stdio.h>

#include<math.h>

int main()

{

int sum=0, i, n,x;

scanf("%d",&n);

fflush(stdin);

scanf("%d",&x);

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

{

sum+= ((pow(-1,i+1))\*(pow(x,i)));

}

printf("\nOUTPUT:\n%d",sum);

return 0;

}

## **Source 69: reverse day so(đảo ngược)**

#include<stdio.h>

int main(){

int i,j;

long n;

do

scanf("%d",&n);

while (n<0);

long rev=0;

while (n!=0){

rev=rev\*10+n%10;

n/=10;

}

n=rev;

printf("%d",n);

}

## **Source 70: square odd number(số lẻ vuông)**

#include<stdio.h>

int main(){

int size;

scanf("%d",&size);

int i,Arr[size];

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

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

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

if(Arr[i]%2==1&& Arr[i]>0) {

printf("%d ",Arr[i]\*Arr[i]);

}

printf("\n");

return 0;

}

## **Source 71: in ra vi tri co tong bang` mot so nao` do**

#include<stdio.h>

int main (){

int size;

scanf("%d",&size);

int i,j,Arr[size];

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

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

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

for(j=size-1;j>=0;j--)

{

if(Arr[i]+Arr[j]==5)

printf("Sum = 5 at position %d and %d\n",i,j);

}

return 0;

}

## **Source 72: Diện tích và chu vi hình chữ nhật**

Diện tích hình chữ nhật = chiều dài x chiều rộng

Chu vi hình chữ nhật = 2 x (chiều dài + chiều rộng)

#include <stdio.h>

int main() {

float chieu\_dai = 5.0;

float chieu\_rong = 3.0;

float dien\_tich, chu\_vi;

dien\_tich = chieu\_dai \* chieu\_rong;

chu\_vi = 2 \* (chieu\_dai + chieu\_rong);

printf("Chieu dai = %f\n", chieu\_dai);

printf("Chieu rong = %f\n", chieu\_rong);

printf("Dien tich = %f\n", dien\_tich);

printf("Chu vi = %f\n", chu\_vi);

return 0;

}

## **Source 73: Diện tích và chu vi tam giác:**

//Di?n tích tam giác = (dáy x chi?u cao) / 2

//

//Chu vi tam giác = c?nh a + c?nh b + c?nh c

#include <stdio.h>

int main() {

float day = 5.0;

float chieu\_cao = 3.0;

float canh\_a = 4.0;

float canh\_b = 3.0;

float canh\_c = 5.0;

float dien\_tich, chu\_vi;

dien\_tich = (day \* chieu\_cao) / 2;

chu\_vi = canh\_a + canh\_b + canh\_c;

printf("Day = %f\n", day);

printf("Chieu cao = %f\n", chieu\_cao);

printf("Canh a = %f\n", canh\_a);

printf("Canh b = %f\n", canh\_b);

printf("Canh c = %f\n", canh\_c);

printf("Dien tich = %f\n", dien\_tich);

printf("Chu vi = %f\n", chu\_vi);

return 0;

}

## **Source 74: Diện tích và chu vi hình tròn:**

- Dien tích hình tròn = pi x bán kính x bán kính

- Chu vi hình tròn = 2 x pi x bán kính \*/

#include <stdio.h>

#define PI 3.14159265358979323846

int main() {

float ban\_kinh = 3.0;

float dien\_tich, chu\_vi;

dien\_tich = PI \* ban\_kinh \* ban\_kinh;

chu\_vi = 2 \* PI \* ban\_kinh;

printf("Ban kinh = %f\n", ban\_kinh);

printf("Dien tich = %f\n", dien\_tich);

printf("Chu vi = %f\n", chu\_vi);

return 0;

}

## **Source 75: Tính giai thừa F1(n) = n!**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main(){

int n;

scanf("%d", &n);

int F1 = 1, i;

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

F1 \*= i;

}

printf("OUTPUT:\n");

printf("%d\n", F1);

return 0;

}

TEST CASE:

A black background with white text

Description automatically generated

## **Source 76: Tinh tong phan so F2(n) = 1 + 2 + 3 + ... + n**

void SumFraction(int n){

int F2 = 0, i;

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

F2 += i;

}

printf("F2(%d) = %d\n", n, F2);

}

// Tinh F1(n) + F2(n) = m

void SumF1F2(int n, int m){

int tempF1, i;

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

tempF1 = 1;

int j;

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

tempF1 \*= j;

}

if(tempF1 + (int)(i \* (i + 1) / 2.0) == m){

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

break;

}

}

}

## **Source 77: Nhap so nguyen n**

int Input(){

int n;

printf("Enter an int num:");

scanf("%d", &n);

return n;

}

//Tinh n^m

void Tinhsomu(int n, int m){

int nm = 1, i;

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

nm \*= n;

}

printf("%d^%d = %d\n", n, m, nm);

}

int main(){

int n = Input();

FactorNumber(n);

SumFraction(n);

int m;

printf("Enter an int m:");

scanf("%d", &m);

SumF1F2(n, m);

Tinhsomu(n, m);

return 0;

}

## **Source 78: In ra cac so chan o vi tri le theo thu tu giam dan coi v? tri tri dau la vi tri 1**

int main() {

int n, a[20], b[10], k = 0;

int i, j;

printf("Nhap so phan tu cua mang (5 <= n <= 20): ");

scanf("%d", &n);

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

printf("Nhap phan tu thu %d: ", i+1);

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

if (i % 2 != 0 && a[i] % 2 == 0) { // kiem tra vi tri le va gia tri chan

b[k] = a[i];

k++;

}

}

// sắp xếp mang b theo thu tu giam dan

for (i = 0; i < k-1; i++) {

for (j = i+1; j < k; j++) {

if (b[i] < b[j]) {

int temp = b[i];

b[i] = b[j];

b[j] = temp;

}

}

}

// in ket qua

printf("Cac so chan o vi tri le theo thu tu giam dan: ");

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

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

}

printf("\n");

return 0;

}

## **Source 79: Nhập chuỗi và in ra từ dài nhất và ngắn nhất**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main(){

char str[100], word[100], largest[100], smallest[100];

int i, j, len, largest\_len = 0, smallest\_len = 100;

fgets(str, 100, stdin);

j = 0;

for(i=0; i<=strlen(str); i++){

if(str[i]==' ' || str[i]=='\0'){

word[j] = '\0';

len = strlen(word);

if(len > largest\_len){

largest\_len = len;

strcpy(largest, word);

}

if(len < smallest\_len){

smallest\_len = len;

strcpy(smallest, word);

}

j = 0;

}

else{

word[j] = str[i];

j++;

}

}

printf("OUTPUT:\n");

printf("%s\n", largest); //The largest word

printf("%s\n", smallest); //The smallest word

return 0;

}

TEST CASE:

A black screen with white text

Description automatically generated

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <ctype.h>

int main() {

system("cls");

//INPUT - @STUDENT:ADD YOUR CODE FOR INPUT HERE:

int i,count = 0;

char s[100];

gets(s);

for(i =0 ; s[i]!='\0';i++){

if(s[i]>='0'&&s[i]<='9'&&(int)s[i]%2==0) count++;

}

// Fixed Do not edit anything here.

printf("\nOUTPUT:\n");

//@STUDENT: WRITE YOUR OUTPUT HERE:

printf("%d",count);

//--FIXED PART - DO NOT EDIT ANY THINGS HERE

printf("\n");

system ("pause");

return(0);

}

## **Source 80: dao nguoc chuoi string**

#include <stdio.h>

// Function to calculate the length of a string

int string\_length(char str[]) {

int length = 0;

while (str[length] != '\0') {

length++;

}

return length;

}

// Function to reverse a string

void string\_reverse(char str[]) {

int i, j, len;

char ch;

len = string\_length(str);

j = len - 1;

i = 0;

while (i < j) {

ch = str[j];

str[j] = str[i];

str[i] = ch;

i++;

j--;

}

}

int main() {

char n[100]; // Assuming a maximum string length of 100 characters

scanf("%s", n);

string\_reverse(n);

printf("Reversed string: %s\n", n);

return 0;

}

## **Source 81: In hoa chữ cái đầu**

#include <stdio.h>

#include <ctype.h>

int main() {

char sentence[201];

fgets(sentence, sizeof(sentence), stdin);

int i = 0;

int capitalizeNext = 1; // Flag to indicate if the next character should be capitalized

while (sentence[i] != '\0') {

if (isalpha(sentence[i])) {

if (capitalizeNext) {

sentence[i] = toupper(sentence[i]);

capitalizeNext = 0;

} else {

sentence[i] = tolower(sentence[i]);

}

} else if (isspace(sentence[i])) {

capitalizeNext = 1;

}

i++;

}

printf("\nOUTPUT:\n%s", sentence);

return 0;

}

## **Source 82: Ma Trận:**

#include <stdio.h>

#define MAX\_SIZE 100

void addMatrices(int A[][MAX\_SIZE], int B[][MAX\_SIZE], int C[][MAX\_SIZE], int rows, int cols) {

int i,j;

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

for (j = 0; j < cols; j++) {

C[i][j] = A[i][j] + B[i][j];

}

}

}

int main() {

int N, M,i,j;

int A[MAX\_SIZE][MAX\_SIZE];

int B[MAX\_SIZE][MAX\_SIZE];

int C[MAX\_SIZE][MAX\_SIZE];

// Read input

scanf("%d %d", &N, &M);

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

for (j = 0; j < M; j++) {

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

}

}

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

for (j = 0; j < M; j++) {

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

}

}

// Add matrices

addMatrices(A, B, C, N, M);

// Print output

printf("\nOUTPUT:\n");

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

for (j = 0; j < M; j++) {

printf("%d ", C[i][j]);

}

printf("\n");

}

return 0;

}

## **Source 83: Tổng từng số bằng tích từng số**

#include <stdio.h>

// Hàm tính t?ng các ch? s? c?a m?t s? nguyên duong

int sumOfDigits(int num) {

int sum = 0;

while (num > 0) {

sum += num % 10; // L?y ch? s? cu?i cùng và c?ng vào t?ng

num /= 10; // Lo?i b? ch? s? cu?i cùng

}

return sum;

}

// Hàm tích các ch? s? c?a m?t s? nguyên duong

int productOfDigits(int num) {

int product = 1;

while (num > 0) {

product \*= num % 10; // L?y ch? s? cu?i cùng và nhân vào tích

num /= 10; // Lo?i b? ch? s? cu?i cùng

}

return product;

}

int main() {

int number;

printf("Nh?p m?t s? nguyên duong: ");

scanf("%d", &number);

int sum = sumOfDigits(number);

int product = productOfDigits(number);

if (sum == product) {

printf("yes");

} else {

printf("no");

}

return 0;

}

## **Source 84: Sắp xếp số lẻ trước chẵn sau giảm dần**

#include <stdio.h>

#include <math.h>

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

int main() {

int n;

scanf("%d", &n);

int i, a[n];

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

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

}

sort(a, n);

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

if (a[i] % 2 == 0) {

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

}

}

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

if (a[i] % 2) {

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

}

}

return(0);

}

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

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

int i, j;

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

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

if (a[j + 1] > a[j]) {

swap(&a[j + 1], &a[j]);

}

}

}

}

## **Source 85: Xóa hết cách kí tự số**

#include <stdio.h>

#include <ctype.h>

int main() {

char input[101];

int i;

fgets(input, sizeof(input), stdin);

for (i = 0; input[i] != '\0'; i++) {

if (isalpha(input[i]) || isspace(input[i])) {

printf("%c", input[i]);

}

}

return 0;

}

## **Source 86: in lại các phần tử trong mảng 1 lần duy nhất**

#include <stdio.h>

int main() {

int n,i,j;

scanf("%d", &n);

int arr[n];

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

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

}

int unique\_arr[n], unique\_count = 0;

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

int is\_duplicate = 0;

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

if (arr[i] == arr[j]) {

is\_duplicate = 1;

break;

}

}

if (!is\_duplicate) {

unique\_arr[unique\_count] = arr[i];

unique\_count++;

}

}

printf("OUTPUT: \n");

for (i = 0; i < unique\_count; i++) {

if (i == unique\_count - 1) {

printf("%d\n", unique\_arr[i]);

} else {

printf("%d ", unique\_arr[i]);

}

}

return 0;

}

## **Source 87: Tìm tên dựa vào chữ cái đầu tiên**

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

getchar();

char s[100];

scanf("%[^\n]", s);

fflush(stdin);

char find;

scanf("%c", &find);

char a[100][100];

int cnt = 0;

char\* token = strtok(s, " ");

while (token != NULL) {

strcpy(a[cnt++], token);

token = strtok(NULL, " ");

}

int i;

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

if (a[i][0] == find) {

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

}

}

return(0);

}

## **Source 88: Tong cac so le trong mang**

#include <stdio.h>

int main()

{

int n;

scanf("%d",&n);

int arr[n],i;

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

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

int sum=0;

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

if(arr[i]%2!=0)

sum+=arr[i];

printf("OUTPUT:\n%d",sum);

}

## **Source 89: In hình kim cương số nguyên tố**

#include <stdio.h>

#include <math.h>

int main() {

int rows;

scanf("%d", &rows);

int prime[1000], cnt = 0;

int i, j;

for (i = 0; i < 1000; i++) {

if (isPrime(i)) {

prime[cnt] = i;

++cnt;

}

}

int number = 0;

for (i = rows; i > 0; i--) {

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

printf("%d ", prime[number]);

++number;

}

printf("\n");

}

return(0);

}

int isPrime(int n) {

int i;

if (n < 2) {

return 0;

}

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

if (n % i == 0) {

return 0;

}

}

return 1;

}

## **Source 90: in số lớn nhất của đường chéo chính trong ma trận**

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int a[n][n];

int i, j;

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

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

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

}

getchar();

}

int max = -1e6 + 1;

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

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

if (i == j) {

if (a[i][j] > max) {

max = a[i][j];

}

}

}

}

printf("%d", max);

return(0);

}

## **Source 91: In dãy fibonasi**

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int i;

long long fibo[n];

fibo[0] = 0; fibo[1] = 1;

printf("0 1 ");

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

fibo[i] = fibo[i - 1] + fibo[i - 2];

printf("%lld ", fibo[i]);

}

return(0);

}

## **Source 92: Đếm số lượng chữ số chẵn và lẻ**

#include <stdio.h>

int main() {

int n;

printf("Nh?p m?t s? nguyên duong: ");

scanf("%d", &n);

int evenCount = 0; // S? lu?ng ch? s? ch?n

int oddCount = 0; // S? lu?ng ch? s? l?

// Ð?m ch? s? ch?n và l?

int digit;

while (n > 0) {

digit = n % 10; // L?y ch? s? cu?i cùng

if (digit % 2 == 0) {

evenCount++;

} else {

oddCount++;

}

n /= 10; // Lo?i b? ch? s? cu?i cùng

}

printf("S? lu?ng ch? s? ch?n: %d\n", evenCount);

printf("S? lu?ng ch? s? l?: %d\n", oddCount);

return 0;

}

## **Source 93: In ra tất cả các chữ in thường**

#include <stdio.h>

#include <ctype.h> // Ð? s? d?ng hàm islower

int main() {

int i;

char inputString[100]; // Khai báo m?ng kí t? d? luu chu?i

printf("Nh?p m?t chu?i kí t?: ");

fgets(inputString, sizeof(inputString), stdin); // Nh?n chu?i t? ngu?i dùng

printf("Các kí t? vi?t thu?ng trong chu?i:\n");

for (i = 0; inputString[i] != '\0'; i++) {

if (islower(inputString[i])) {

printf("%c ", inputString[i]);

}

}

printf("\n");

return 0;

}

## **Source 93: Chuyển tất cả về chữ cái thường**

#include <stdio.h>

#include <ctype.h> // Để sử dụng hàm tolower

int main() {

char inputString[100]; // Khai báo mảng kí tự để lưu chuỗi

printf("Nhập một chuỗi kí tự: ");

fgets(inputString, sizeof(inputString), stdin); // Nhận chuỗi từ người dùng

printf("Chuỗi sau khi chuyển thành chữ in thường: ");

for (int i = 0; inputString[i] != '\0'; i++) {

char lowercaseChar = tolower(inputString[i]);

printf("%c", lowercaseChar);

}

printf("\n");

return 0;

}

## **Source 93: Chuyển tất cả về chữ cái hoa**

#include <stdio.h>

#include <ctype.h> // Để sử dụng hàm toupper

int main() {

char inputString[100]; // Khai báo mảng kí tự để lưu chuỗi

printf("Nhập một chuỗi kí tự: ");

fgets(inputString, sizeof(inputString), stdin); // Nhận chuỗi từ người dùng

printf("Chuỗi sau khi chuyển thành chữ in hoa: ");

for (int i = 0; inputString[i] != '\0'; i++) {

char uppercaseChar = toupper(inputString[i]);

printf("%c", uppercaseChar);

}

printf("\n");

return 0;

}

## **Source 94: Hoán đổi số chẵn lớn nhất với số chắn nhỏ nhất**

#include <stdio.h>

// Hàm tìm s? ch?n l?n nh?t trong m?ng

int findMaxEven(int arr[], int n) {

int i;

int maxEven = -1; // Gi? s? không có s? ch?n trong m?ng

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

if (arr[i] % 2 == 0 && (maxEven == -1 || arr[i] > maxEven)) {

maxEven = arr[i];

}

}

return maxEven;

}

// Hàm tìm s? ch?n nh? nh?t trong m?ng

int findMinEven(int arr[], int n) {

int i;

int minEven = -1; // Gi? s? không có s? ch?n trong m?ng

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

if (arr[i] % 2 == 0 && (minEven == -1 || arr[i] < minEven)) {

minEven = arr[i];

}

}

return minEven;

}

int main() {

int n, i;

printf("Nh?p s? lu?ng ph?n t? trong m?ng: ");

scanf("%d", &n);

int arr[100];

printf("Nh?p các ph?n t? c?a m?ng:\n");

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

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

}

int maxEven = findMaxEven(arr, n);

int minEven = findMinEven(arr, n);

if (maxEven != -1 && minEven != -1) {

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

if (arr[i] == maxEven) {

arr[i] = minEven;

} else if (arr[i] == minEven) {

arr[i] = maxEven;

}

}

printf("M?ng sau khi hoán d?i s? ch?n l?n nh?t và nh? nh?t:\n");

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

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

}

printf("\n");

} else {

printf("Không tìm th?y s? ch?n trong m?ng.\n");

}

return 0;

}

## **Source 95: In hoa 2 chữ cái đầu mỗi từ**

#include <stdio.h>

#include <ctype.h> // Ð? s? d?ng hàm toupper

void capitalizeFirstTwo(char str[]) {

int i = 0;

while (str[i] != '\0') {

// B?t d?u m?t t? m?i

while (str[i] != '\0' && !isalpha(str[i])) {

putchar(str[i]);

i++;

}

// Vi?t hoa kí t? d?u tiên n?u là ch?

if (isalpha(str[i])) {

putchar(toupper(str[i]));

i++;

// Vi?t hoa kí t? th? hai n?u có

if (isalpha(str[i])) {

putchar(toupper(str[i]));

i++;

}

}

// Copy ph?n còn l?i c?a t?

while (str[i] != '\0' && !isspace(str[i])) {

putchar(str[i]);

i++;

}

}

}

int main() {

char inputString[100]; // Khai báo m?ng kí t? d? luu chu?i

printf("Nh?p m?t chu?i: ");

fgets(inputString, sizeof(inputString), stdin); // Nh?n chu?i t? ngu?i dùng

printf("Chu?i sau khi vi?t hoa hai ký t? d?u c?a m?i t?: ");

capitalizeFirstTwo(inputString);

return 0;

}

## **Source 96 : in ra các phần tử bị lặp lại trong mảng.**

#include <stdio.h>

void findDuplicates(int arr[], int n) {

printf("Các phần tử bị lặp lại trong mảng: ");

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

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

if (arr[i] == arr[j]) {

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

break; // Để tránh in phần tử bị lặp lại nhiều lần

}

}

}

printf("\n");

}

int main() {

int n;

printf("Nhập số lượng phần tử trong mảng: ");

scanf("%d", &n);

int arr[100];

printf("Nhập các phần tử của mảng:\n");

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

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

}

findDuplicates(arr, n); // In ra các phần tử bị lặp lại trong mảng

return 0;

}

## **Source 97 : Đếm có bao nhiêu ký tự nhập vào trong srting**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int countSubstring(char str[], char sub[]) {

int count = 0;

int i;

int len1 = strlen(str);

int len2 = strlen(sub);

for (i = 0; i <= len1 - len2; i++) {

int j;

for (j = 0; j < len2; j++) {

if (tolower(str[i+j]) != tolower(sub[j]))

break;

}

if (j == len2) {

count++;

j = 0;

}

}

return count;

}

int main() {

char input[100];

printf("Please enter strings: ");

fgets(input, sizeof(input), stdin);

input[strcspn(input, "\n")] = '\0'; // remove trailing newline

char k[100];

printf("Please enter string k: ");

fgets(k, sizeof(k), stdin);

k[strcspn(k, "\n")] = '\0'; // remove trailing newline

int count = countSubstring(input, k);

printf("%d", count);

return 0;

}

## **Source 98: Vẽ hình chữ A tam giác**

#include <stdio.h>

int main() {

int row;

printf("Enter the number of rows: ");

scanf("%d", &row);

int i,j,k;

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

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

printf("%c", 'A' + j - 1);

}

for ( k = i - 1; k >= 1; k--) {

printf("%c", 'A' + k - 1);

}

printf("\n");

}

return 0;

}

## **Source 99: sắp xếp tang dần**

#include <stdio.h>

#include <stdlib.h>

int main() {

int n;

scanf("%d", &n);

float a[n];

int i;

for(i=0; i<n; i++) scanf("%f", &a[i]);

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

int j;

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

if (a[j] > a[j+1]) {

int temp = a[j];

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

a[j+1] = temp;

}

}

}

printf("OUTPUT:\n");

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

printf("%.3f ", a[i]);

}

return 0;

}

## **Source 100: bài sắp xếp chẵn trước lẻ sau**

#include <stdio.h>

void swap(int \*a, int \*b); // function prototype

int main() {

int n,i,j;

printf("Please enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Please enter %d integers separated by spaces: ", n);

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

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

}

// Sort the array in ascending order with odd numbers first, even numbers later

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

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

if ((arr[i] % 2 != 0 && arr[j] % 2 != 0 && arr[i] > arr[j]) || (arr[i] % 2 != 0 && arr[j] % 2 == 0)) {

swap(&arr[i], &arr[j]);

}

}

}

printf("Sorted array with odd numbers first and even numbers later: ");

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

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

}

printf("\n");

return 0;

}

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

## **Source 80: Chuyển đổi số nguyên thành mã ASCII:**

#include <stdio.h>

int main() {

int number;

printf("Enter an integer: ");

scanf("%d", &number);

printf("ASCII code of %d is %c\n", number, number);

return 0;

}

## Source 81: **Chuyển đổi mã ASCII thành ký tự:**

#include <stdio.h>

int main() {

char character;

printf("Enter an ASCII code: ");

scanf("%c", &character);

printf("Character of ASCII code %c is %d\n", character, character);

return 0;

}

## **Source 82: Chuyển đổi một chuỗi ký tự thành mã ASCII:**

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

printf("Enter a string: ");

scanf("%s", str);

printf("ASCII codes of characters in the string:\n");

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

printf("%c: %d\n", str[i], str[i]);

}

return 0;

}

## Source 83: **Chuyển đổi từng ký tự trong chuỗi thành mã ASCII:**

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

printf("Enter a string: ");

scanf("%s", str);

printf("ASCII codes of characters in the string:\n");

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

printf("%c: %d\n", str[i], str[i]);

}

return 0;

}

## Source 83: Tổng các chữ số trong một số

#include <stdio.h>

#include <math.h>

int main(){

int n;

printf("Nhap so nguyen duong n:");

scanf("%d",&n);

int sum = 0;

while(n>0){

sum+=n%10;

n=n/10;

}

printf("sum=%d",sum);

return 0;

}

## Source 84: giữ 5 phần tử characters ở giữa và xóa những phần tử khác

#include <stdio.h>

#include <string.h>

int main() {

char input[100]; // Gi? s? d? dài t?i da c?a chu?i là 100 ký t?

char output[7]; // Ð? dài t?i da c?a chu?i k?t qu? là 5 ký t? + 1 ký t? k?t thúc chu?i (\0)

int length;

printf("Nh?p chu?i (d? dài t? 5 d?n 20 ký t?): ");

fgets(input, sizeof(input), stdin);

length = strlen(input);

// Xóa ký t? newline n?u t?n t?i

if (input[length - 1] == '\n') {

input[length - 1] = '\0';

length--;

}

if (length >= 5 && length <= 20) {

int start = (length - 5) / 2;

strncpy(output, input + start, 5);

output[5] = '\0'; // Ð?m b?o k?t thúc chu?i

printf("Chu?i sau khi ch? gi? l?i 5 ký t? ? gi?a: %s\n", output);

} else {

printf("Ð? dài c?a chu?i không th?a mãn yêu c?u.\n");

}

return 0;

}

## Source 85: check số symmetric

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

int len, i, is\_symmetric = 1;

int n;

scanf("%d\n",&n); // Chú ý \n ? dây d? tiêu tr? ký t? newline sau khi nh?p n.

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

scanf(" %c", &str[i]); // Ð?c t?ng ký t? và luu vào m?ng str.

}

str[n] = '\0'; // K?t thúc chu?i b?ng ký t? null.

len = strlen(str);

for (i = 0; i < len/2; i++) {

if (str[i] != str[len-i-1]) {

is\_symmetric = 0;

break;

}

}

if (is\_symmetric) {

printf("The string is symmetric");

}

else {

printf("The string is not symmetric");

}

return 0;

}

## Source 86: sắp xếp phần tử lẻ giảm dần và chẵn tăng dần trên 1 dòng output

#include <stdio.h>

void sap\_xep(int a[], int n) {

int i, j;

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

int m = i;

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

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

m = j;

}

int c = a[m];

a[m] = a[i];

a[i] = c;

}

}

void dao\_vi\_tri(int a[], int n) {

int left = 0, right = n - 1;

while (left < right) {

int temp = a[left];

a[left] = a[right];

a[right] = temp;

left += 2; // V? trí s? l? ch? d?i cho s? l?

right -= 2; // V? trí s? ch?n ch? d?i cho s? ch?n

}

}

int main() {

int n, i;

printf("Nhap so phan tu cua mang: ");

scanf("%d", &n);

int arr[n];

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

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

}

int chan[n], le[n];

int c = 0, l = 0;

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

if (arr[i] % 2 == 0)

chan[c++] = arr[i];

else

le[l++] = arr[i];

}

sap\_xep(chan, c);

sap\_xep(le, l);

dao\_vi\_tri(le, l);

dao\_vi\_tri(chan, c);

int idx\_le = 0, idx\_chan = 0;

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

if (arr[i] % 2 == 0)

printf("%d ", chan[idx\_chan++]);

else

printf("%d ", le[idx\_le++]);

}

return 0;

}

## Source 87: sắp xếp các phần tử lẻ tăng dần và chẵn giảm dần

Tương tự như trên

#include <stdio.h>

void sap\_xep(int a[], int n) {

int i, j;

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

int m = i;

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

if (a[j] > a[m])

m = j;

}

int c = a[m];

a[m] = a[i];

a[i] = c;

}

}

void dao\_vi\_tri(int a[], int n) {

int left = 0, right = n - 1;

while (left < right) {

int temp = a[left];

a[left] = a[right];

a[right] = temp;

left += 2; // V? trí s? l? ch? d?i cho s? l?

right -= 2; // V? trí s? ch?n ch? d?i cho s? ch?n

}

}

int main() {

int n, i;

printf("Nhap so phan tu cua mang: ");

scanf("%d", &n);

int arr[n];

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

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

}

int chan[n], le[n];

int c = 0, l = 0;

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

if (arr[i] % 2 == 0)

chan[c++] = arr[i];

else

le[l++] = arr[i];

}

sap\_xep(chan, c);

sap\_xep(le, l);

dao\_vi\_tri(le, l);

dao\_vi\_tri(chan, c);

int idx\_le = 0, idx\_chan = 0;

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

if (arr[i] % 2 == 0)

printf("%d ", chan[c - 1 - idx\_chan++]);

else

printf("%d ", le[idx\_le++]);

}

return 0;

}

## Source88: Tổng 3 số cuối của mảng

#include <stdio.h>

int main() {

int n, i, t = 0;

printf("Nhap so phan tu cua mang: ");

scanf("%d", &n);

int arr[n];

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

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

}

for (i = n - 3; i < n; i++) {

t = t + arr[i];

}

printf("Tong cua ba so cuoi la: %d\n", t);

return 0;

}

## Source 89: đếm số lượng số chẵn và số lẻ

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <ctype.h>

int main() {

system("cls");

//INPUT - @STUDENT:ADD YOUR CODE FOR INPUT HERE:

int i,m=0,n=0;

int a[5];

for(i=0;i<5;i++){

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

if(a[i]%2==0)

m++;

else

n++;

}

// Fixed Do not edit anything here.

printf("\nOUTPUT:\n");

//@STUDENT: WRITE YOUR OUTPUT HERE:

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

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

//--FIXED PART - DO NOT EDIT ANY THINGS HERE

printf("\n");

system ("pause");

return(0);

}

## Source 90: Số chính phương

#include <stdio.h>

#include <math.h>

int main() {

int num;

printf("Nhap mot so nguyen: ");

scanf("%d", &num);

int sqrt\_num = sqrt(num);

if (sqrt\_num \* sqrt\_num == num) {

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

} else {

printf("%d khong phai la so chinh phuong.\n", num);

}

return 0;

}

## Source 91 : In ra các số chính phương trong khoảng từ a->b

#include <stdio.h>

#include <math.h>

int main() {

int A, B;

printf("Nhap khoang A den B: ");

scanf("%d %d", &A, &B);

printf("Cac so chinh phuong tu %d den %d la:\n", A, B);

for (int num = A; num <= B; num++) {

int sqrt\_num = sqrt(num);

if (sqrt\_num \* sqrt\_num == num) {

printf("%d ", num);

}

}

return 0;

}

## Source 92: Tổng các số chính phương trong mảng

#include <stdio.h>

#include <math.h>

int main() {

int n;

printf("Nhap so phan tu cua mang: ");

scanf("%d", &n);

int arr[n];

printf("Nhap cac phan tu cua mang:\n");

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

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

}

int sum = 0;

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

int sqrt\_num = sqrt(arr[i]);

if (sqrt\_num \* sqrt\_num == arr[i]) {

sum += arr[i];

}

}

printf("Tong cac so chinh phuong trong mang la: %d\n", sum);

return 0;

}

 1. (bai tap 8): nhap vao so giay, in ra so nam,thang, ngay, gio,phut, giay (2)

 2. nhap so tien ( nghin dong) co cac menh gia  thong dung 500k, 200k, 100k....1k. in ra so tien theo menh gia sao cho so to tien it nhat(khong dung array) (2)

3. Lam cac bai tap tu a....g, h, i, j ( tương tự với các bài đã chữa của SNT) đối với các số Fibonacy, Palindrome, Armstrong, số may mắn, số hạnh phúc (1.5\*4)

(//1. nhap so nguyen n,

// a. kiem tra xem co phai so nguyen to ko?

//neu phai, in ra no la SNT;

//b. neu khong phai snt, in ra cac uoc so cua no.

// c. in ra tong tat ca cac uoc so cua n

// d. in ra so luong cac cac uoc so cua n

// e. nhap n, in ra cac SNT tu 1 den n

// f. in ra n SNT dau tien

// g. in ra 3 SNT lon hon n

// h. in ra 2 SNT, mot so lon hon va mot so nho hon n  (gan n nhat)

// i. in ra 4 SNT, 2 so lon hon va 2 so nho hon n

// j. nhap n,p, q, in ra p SNT nho hon n va q SNT lon hon n.

// k. in ra cac SNT (co 3, hoac co 4 chu so) ma co tong cac chu so

//chia het cho 5

//  l. nhập n, tìm các số vừa là SNT vừa là Fibonacy nhỏ hơn n

 1. Tính các tổng (4)

a. //Tinh S=1^2+2^2+3^2+...+n^2

b.  33: Tinh S4=1^2+2^3+3^4+...+n^(n+1)

c. Nhap x, n. Tinh S=x^1+x^2+...+x^n

d. Tinh S=1/1+1/1(1+2)+1/(1+2+3)+...+1/(1+2+...+n)

e. Tinh S= 1/1!+1/(2!)+1/(3!)+...+1/(n!)

f.  Tinh S=1-1/2+1/3-1/4+1/5-...+(/-)1/n

g. Nhap n,x. Tinh S=1+x^2/2!+x^4/4!+...+x^(2n)/(2n!)

1. mo rong: nhap mang n phan tu,  sap xep cac phan tu i o vi tri chan ( hoacco gia tri chan a[i])

2.  //a. Sap xep  cac phan tu o vi tri chan tang dan, phan tu o vi tri le giam dan-OK

 //b. Sap xep cac phan tu  co gia tri chan tang dan, gia tri le giam dan-OK

3. // BT20: cach 1 Nhap va sap xep  mang n phan tu,

//dung thuat toan interchange

// sap xep tang dan cac so floating point co 3 chu so thap phan,

// cach nhau boi dau -

 4.  BTVN 24. mo rong 2: nhap array n phan tu,va cac gia tri b,e

 //sap xep gia tri tu phan tu b den phan tu e bat ky trong mang theo thu tu tang dan (ke ca dau mut)

Workshop 4:

1. Sua : // nhap mang n phan tu, nhap a va b,  thay the cac  so truoc va sau snt nam trong khoang a[a] va a[b] bang so 99

2. them

a. Bai 36. Nhap vao mot mang co n phan tu,  them vao phia truoc tat ca  SNT bang so 0 (ok)

b. Them so 99 vao dang truoc 2 phan tu (phan tu can thay k=-2) so voi phan tu thu 3 la   SNT (phan tu goc p=3)

c. //Bai 36-37. Nhap vao mot mang co n phan tu, them vao phia truoc tat ca cac phan tu la SNT bang so 99 (ok)

Workshop 5:

a. bai 40 : xoa tat ca cac SNT

b.  xóa bo phan tu chan thu 2 (  hoac phan tu NT thu 2)

c. Xoa bo 2 (hoac 3)  phan tu dung sau phan tu chan  (hoac phan tu NT) thu 2 (k=1)

## **Source 93 : Nguyen am thanh chu hoa phu am thanh chu thuong**

#include <stdio.h>

int main() {

char s[1000];

gets(s);

int i;

for (i = 0; i < strlen(s); i++) {

if (isupper(s[i])) {

s[i] +=32;

}if (s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u') {

s[i] -= 32;

}

}

puts(s);

}

## **Source 93 : Kiem tra co la pow cua 2**

#include <stdio.h>

int power(int n) {

if (n == 0) return 0;

int i,count = 0;

for (i = 1; i > 0;i = i<<1) {

if (i == n) {

return count;

}

if (i > n) {

return 0;

}

count++;

}

return 0;

}

int main() {

int n;

scanf("%d", &n);

if (power(n) != 0) {

printf("%d", power(n));

} else {

printf("%d is not a power of 2",n);

}

}