- cách reset mảng được khai báo ở ngoài hàm main:

int dt[10000];

int main(){

memset(dt, 0, sizeof(dt));

}

- Khai báo mảng: Kiểu dữ liệu + tên mảng + [Kích thước mảng]

VD: int n[5] sẽ khai báo 1 mảng gồm 5 ô số nguyên

+Mảng có độ lớn n thì các ô nhớ sẽ chạy từ 0 đến n – 1;

int a[100]; //khai bao 1 mang so nguyen co 100 phan tu

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

printf("nhap gia tri cho phan tu thu %d :", i + 1);

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

}

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

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

}

printf("\n");

for(int i = 4; i >= 0; i--){

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

}

- khai báo mảng, số lượng phần tử của mảng được nhập từ bàn phím;

int n;

printf("nhap so luong phan tu cua mang : ");

scanf("%d", &n);

int a[n];

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

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

}

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

- khai báo mảng kiểu thủ công (viết thẳng vào code)

int a[5] = {2, 3, 4, 1, 6};

for(int i = 0; i < 5; i++) printf("%d", a[i]);

return 0;

- Nếu khai báo kiểu thủ công, thì nếu độ lớn của mảng là 100, mà ta chỉ nhập 5 giá trị đầu, thì 95 giá trị còn lại đều bằng 0;

int a[100] = {2, 3, 4, 1, 6};

for(int i = 0; i < 100; i++) printf("%d ", a[i]);

- Khai báo 1 mảng toàn số 0;

int a[100] = {0};

- nhập mảng có tên a như một đối số vào trong hàm

\* lưu ý: lúc gọi hàm, ở vị trí của mảng, chỉ cần ghi a vào. VD: int a[100]; nhap(a, n);

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

printf("nhap cac phan tu cua mang: ");

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

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

}

}

int main() {

int n, a[1000];

scanf("%d", &n);

nhap(a, n);

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

return 0;

}

- Cách sử dụng hàm qsort;

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <ctype.h>

int tcs(int n){

int sum = 0;

while(n != 0){

sum += (n%10);

n /= 10;

}

return sum;

}

int cmp(const void \*a, const void \*b){

// -1 ben trai, 1 ben phai

//khi muon a dung truoc thi return -1 (mot gia tri am)

//khi muon b dung truoc thi return 1 (mot gia tri duong)

int \*x = (int\*)a;

int \*y = (int\*)b;

if(\*x%2==0 && \*y%2==1) return -1;

if(\*x%2==1 && \*y%2==0) return 1;

if(\*x%2==0 && \*y%2==0) return \*y - \*x;

return \*x - \*y;

}

int main(){

int n; scanf("%d", &n);

int a[n];

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

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

}

qsort(a, n, sizeof(int), cmp);

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

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

}

}