

BT1.CPP

LOẠI BỎ TRÙNG NHAU

Viết chương trình nhập vào mảng A gồm n phần tử số nguyên ($2 < n < 100$). Thực hiện loại bỏ các phần tử đã xuất hiện và in ra mảng kết quả.

Input

Dòng thứ nhất là số phần tử của mảng A.

Dòng thứ hai là các phần tử của mảng A

Output

Ghi ra kết quả trên một dòng

Ví dụ:

Input	Output
7 1 5 10 10 5 2 3	1 5 10 2 3

BT1.CPP

```
1  #include<stdio.h>
2  #define MAX 100
3  void botrungnhau(int a[],int n){
4      int i,b[MAX] = {0};
5      for(i=0;i<n;i++){
6          scanf("%d",&a[i]);
7          b[a[i]]++;
8      }
9      for(i=0;i<n;i++){
10         if (b[a[i]]>=1){
11             printf("%3d ",a[i]);
12             b[a[i]] = 0;
13         }
14     }
15 }
16 int main(){
17     int n,a[MAX];
18     scanf("%d",&n);
19     botrungnhau(a,n);
20     return 0;
21 }
```

BT2.CPP

SẮP XẾP CHỌN

Hãy thực hiện thuật toán sắp xếp chọn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

Dữ liệu vào: Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

Kết quả: Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

Ví dụ:

Input	Output
4	Bước 1: 2 7 3 5
5 7 3 2	Bước 2: 2 3 7 5
	Bước 3: 2 3 5 7

BT2.CPP

```
1  #include <stdio.h>
2  #define MAX 100
3  void swap(int *a, int *b ){
4      int temp =*b;
5      *b=*a;
6      *a=temp;
7  }
8  void xuat(int a[], int n, int i){
9      printf("Buoc %d:",i);
10     for (i=0; i < n; i++){
11         printf(" %d", a[i]);
12     }
13     printf("\n");
14 }
```

```
15 void snchon(int a[],int n){
16     int i,j,min;
17     for(i=0;i<n-1;i++){
18         min=i;
19         for(j=i+1;j<n;j++){
20             if(a[j]<a[min]){
21                 min=j;
22             }
23             if(i!=min){
24                 swap(&a[min],&a[i]);
25             }
26             xuat(a,n,i+1);printf("\n");
27         }
28     }
29 }
30 int main(){
31     int n,i,a[MAX];
32     scanf("%d",&n);
33     for(i=0;i<n;i++){
34         scanf("%d",&a[i]);
35     }
36     snchon(a,n);
37     return 0;
38 }
```

BT3.CPP

SẮP XẾP CHÈN

Hãy thực hiện thuật toán sắp xếp chèn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

Dữ liệu vào: Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

Kết quả: Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

Ví dụ:

Input	Output
4	Buoc 0: 5
5 7 3 2	Buoc 1: 5 7
	Buoc 2: 3 5 7
	Buoc 3: 2 3 5 7

BT3.CPP

```
1  #include <stdio.h>
2  #define MAX 100
3  void xuatkq(int a[],int n,int i){
4      int j;
5      printf("Buoc %d:",i);
6      for(j=0;j<=i;j++){
7          printf("%3d",a[j]);
8      }
9      printf("\n");
10 }
11 void sxchen(int a[],int n){
12     int i,j,x;
13     printf("Buoc 0:%3d\n",a[0]);
14     for(i=1;i<n;i++){
15         x=a[i];
16         j=i-1;
17         while(x<a[j]&& j>=0){
18             a[j+1]=a[j];
19             j--;
20         }
21         a[j+1]=x;
22         xuatkq(a,n,i);
23     }
24 }
```

```
25 int main(){
26     int n,i,a[MAX];
27     scanf("%d",&n);
28     for(i=0;i<n;i++){
29         scanf("%d",&a[i]);
30     }
31     sxchen(a,n);
32     return 0;
33 }
```

BT4.CPP

VỪA NGUYÊN TỐ VỪA THUẬN NGHỊCH

Viết chương trình liệt kê các số vừa nguyên tố, vừa thuận nghịch trong đoạn $[a,b]$ với a,b là các số nguyên dương thỏa mãn $10 < a < b < 10^6$.

Input

Dòng đầu ghi số bộ test. Mỗi bộ test là một cặp số a,b .

Output

Với mỗi bộ test, ghi lần lượt các số thỏa mãn, mỗi số cách nhau một khoảng trống, mỗi dòng ghi đúng 10 số. Khi hết một test thì bỏ trống một dòng trước khi ghi kết quả test tiếp theo.

Ví dụ

Input	Output
2	101 131 151 181 191 313 353 373 383 727
20 1234	757 787 797 919 929
123 140	131

BT4.CPP

```
1  #include<stdio.h>
2  #include<math.h>
3  #include<string.h>
4  int SNT(int n){
5      if ( n < 2) return 0;
6      if ( n==2 || n==3) return 1;
7      for (int i=2 ; i <= sqrt(n); i++){
8          if ( n %i ==0) return 0;
9
10         }
11     return 1;
12
13     }
14 int TN(int n){
15     int m =n;
16     int k =0;
17     while(m >0){
18         k = k*10+ m%10;
19         m = m/10;
20     }
21     if ( k == n) return 1;
22     return 0;
23 }
```

```
24 main(){
25     int t;
26     scanf("%d",&t);
27     while(t--){
28         int a,b;
29         scanf("%d%d",&a,&b);
30         int test =0;
31         for ( int i = a; i <= b ; i++){
32             if ( test == 10){
33                 printf("\n");test =0;
34             }
35             if (SNT(i) ==1 && TN(i)){
36                 printf("%d ",i);
37                 test++;
38             }
39
40         }
41         printf("\n");
42     }
```


BT5.CPP

MẢNG ĐỐI XỨNG

Nhập một dãy số nguyên có n phần tử (n không quá 100, các phần tử trong dãy không quá 10^9). Hãy viết chương trình kiểm tra xem dãy có phải đối xứng hay không. Nếu đúng in ra YES, nếu sai in ra NO.

Input: Dòng đầu ghi số bộ test, mỗi bộ test gồm hai dòng. Dòng đầu là số phần tử của dãy, dòng sau ghi ra dãy đó, mỗi số cách nhau một khoảng trống.

Output: In ra kết quả kiểm tra.

Ví dụ:

Input	Ouput
2	YES
4	NO
1 4 4 1	
5	
1 5 5 5 3	

BT5.CPP

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  void nhap(int *arr, int *n) {
4      *n = 0;
5      while(*n <= 0) {
6          scanf("%d", n);
7      }
8      int i;
9      for(i = 0; i < *n; i++) {
10         scanf("%d", &arr[i]);
11     }
12 }
13 bool ktra(int* arr, int n) {
14     int i;
15     for(i = 0; i <= n/2; i++) {
16         if(arr[i] != arr[n - 1 - i]) {
17             return false;
18         }
19     }
20     return true;
21 }
```

```
22 int main() {
23     int n;
24     int arr[100];
25     int k;
26     scanf("%d", &k);
27     for(int i = 0; i < k ; i++) {
28         nhap(arr, &n);
29         if(ktra(arr, n)) {
30             printf("YES\n");
31         } else {
32             printf("NO\n");
33         }
34     }
35     return 0;
36 }
```

BT6.CPP

```
1  #include <stdio.h>
2  main()
3  {
4      char str[21];
5      printf("Nhap vao mot chuoì: ");
6      //Nên sử dụng nhập chuỗi với hàm gets()
7      gets(str);
8      printf("Bạn vừa nhập vao chuỗi: ");
9      puts(str);
10     fflush(stdin);
11     getchar();
12 }
```

BT7.CPP

```
1  #include <stdio.h>
2  main()
3  {
4      char str1[21]="Hello ";
5      char str2[21]={'W','o','r','l','d','\0'};
6      char *ptr;
7
8      printf("Hai chuoi truoc khi goi ham la:\n");
9      printf("\tstr1: \"%s\"\n", str1);
10     printf("\tstr2: \"%s\"\n", str2);
11
12     //ham noi chuoi str2 vao chuoi str1
13     ptr = strcat(str1, str2);
14
15     printf("\nHai chuoi sau khi goi ham la:\n");
16     printf("\tstr1: \"%s\"\n", str1);
17     printf("\tstr2: \"%s\"\n", str2);
18
19     printf("\nptr=%d, str1=%d\n", ptr, str1);
20
21     fflush(stdin);
22     getchar();
23 }
```

BT8.CPP

```
1  #include <stdio.h>
2  main()
3  {
4      char str[21], ch, *pos;
5      printf("Nhap vao mot chuoi: ");
6      gets(str);
7      printf("Nhap mot ky tu: ");
8      scanf("%c", &ch);
9
10     //ham tim kiem ky tu ch trong chuoi str
11     pos = strchr(str, ch);
12
13     if(pos==NULL)
14         printf("Khong co ky tu '%c' trong chuoi \"%s\"", ch, str);
15     else
16         printf("Ky tu '%c' nam trong chuoi \"%s\" o vi tri thu %d", ch, str, pos-str);
17     fflush(stdin);
18     getchar();
19 }
```

BT9.CPP

```
1  #include <stdio.h>
2  main()
3  {
4      char str1[21]="Hello ";
5      char str2[21]={'W','o','r','l','d','\0'};
6      char *ptr;
7
8      printf("Hai chuoai truoc khi goi ham la:\n");
9      printf("\tstr1: \"%s\"\n", str1);
10     printf("\tstr2: \"%s\"\n", str2);
11
12     //Ham copy chuoai str2 vao chuoai str1
13     ptr = strcpy(str1, str2);
14
15     printf("\nHai chuoai sau khi goi ham la:\n");
16     printf("\tstr1: \"%s\"\n", str1);
17     printf("\tstr2: \"%s\"\n", str2);
18
19     printf("\nptr=%d, str1=%d\n", ptr, str1);
20
21     fflush(stdin);
22     getchar();
23 }
```

BT10.CPP

```
1  #include <stdio.h>
2  main()
3  {
4      char str[21];
5      printf("Nhap vao mot chuoi: ");
6      gets(str);
7      printf("Ban vua nhap vao chuoi co %d ky tu", strlen(str));
8      fflush(stdin);
9      getchar();
10 }
```