

```
1: #include<stdio.h>
2: #include<conio.h>
3: #include <stdlib.h>
4: //Nhap mang 1 chieu:
5: void NhapMang(int *_a, int _n)
6: {int i;
7:     for (i=0;i<_n;i++)
8:     {
9:         printf("\n Nhap phan tu
%d: ",i);
10:         scanf("%d",_a+i);
11:     }
12: }
13: //Xuat mang 1 chieu:
14: void XuatMang(int *_a, int _n)
15: {int i;
16:     for (i=0;i<_n;i++)
17:         printf("\t%d",*(a+i));
18: }
19: // Kiem tra so nt

20: int kt_snt(int _n)
21: {
22:     int dem=0,i;
23:     for(i=1;i<=_n;i++)
24:         if(_n%i==0)
25:             dem++;
26:     if (dem==2) return 1;
27:     return 0;
28: }
29: // Kiem tra so chinh phuong
```

```
30: int kt_scp(int n)
31: {
32:     int i;
33:     for(i=1; i<=n; i++)
34:         if(i*i==n)
35:             return 1;
36:     return 0;
37: }
38: // Chương trình in ra các số nt
39: void In_nto(int *a, int n)
40: {
41:     int i;
42:     for(i=0; i<n; i++)
43:         if(kt_snt(*(a+i))==1)
44:             printf("\t%d", *(a+i));
45: }
46: void In_cp(int *a, int n)
47: {
48:     int i;
49:     for(i=0; i<n; i++)
50:         if(kt_scp(*(a+i))==1)
51:             printf("\t%d", *(a+i));
52: }
53: //
54: int max_nt(int *a, int n)
55: {
56:     int max=-1, i;
57:     for(i=0; i<n; i++)
58:         if(kt_snt(*(a+i))==1)
59:         {
60:             max=*(a+i);
61:             break;
```

```
62:     }
63:     for (i=0; i<_n; i++)
64:         if (kt_snt (* (_a+i) ) == 1  &&
            * (_a+i) > max)
65:             max = * (_a+i) ;
66:     return max;
67: }
68: // Tim so chinh phuong chan max
69: int max_cp (int *_a, int _n)
70: {
71:     int max = -1, i;
72:     for (i=0; i<_n; i++)
73:         if ( (kt_scp (* (_a+i) ) == 1)
            && (* (_a+i) % 2 == 0) )
74:         {
75:             max = * (_a+i) ;
76:             break;
77:         }
78:     for (i=0; i<_n; i++)
79:         if ( (kt_scp (* (_a+i) ) == 1)
            && (* (_a+i) % 2 == 0)  && (* (_a+i) > max) )
80:             max = * (_a+i) ;
81:     return max;
82: }
83: // In ra vi tri cua so co gia tri bang X
84: void In_vt_X (int *_a, int _n, int _X)
85: {
86:     int i;
87:     for (i=0; i<_n; i++)
88:         if (* (_a+i) == _X)
89:         {
```

```
90:         printf("\t %d", i);
91:     }
92: }
93: // In ra vi tri dau tien
94: int In_vtdau_X(int *_a, int _n, int
X)
95: {
96:     int i, vt;
97:     for (i=0; i<_n; i++)
98:         if (*(_a+i) == X)
99:         {
100:             vt=i;
101:             break;
102:         }
103:     return vt;
104: }
105: // In ra vi tri dau tien
106: int In_vtcuoi_X(int *_a, int _n, int
X)
107: {
108:     int i, vt;
109:     for (i=_n-1; i>=0; i--)
110:         if (*(_a+i) == X)
111:         {
112:             vt=i;
113:             break;
114:         }
115:     return vt;
116: }
117: // Chuong trinh chinh
118: int main()
119: {
```

```
120:     int *a,n,i,k,giatri;
121:     do // nhap so phan tu >=1 Nho
        hon 50
122:     {
123:         printf("\n Nhap vao so phan tu
        mang");
124:         scanf("%d",&n);
125:     }
126:     while (n<1 || n>50);
127:
128:     a=(int
        *)malloc(n*sizeof(int)); // C
        phat bo nho dong
129:     NhapMang(a,n);
130:     printf("\n Mang vua nhap
        la:"); // Kiem tra xem nhap da
131:     XuatMang(a,n); // dung hay
        chua
132:     giatri=max_cp(a,n);
133:     if(giatri==-1)
134:         printf("\n Khong co so cp");
135:     else
136:     {
137:         // Cac so cp trong mang la
138:         printf("\n Cac so cp trong
        mang");
139:         In_cp(a,n);
140:         printf("\n So cp chan max la:
        %5d",giatri);
141:         printf("\n Vi tri so cp chan
        max ");
142:         In_vt_X(a,n,giatri);
```

```
143:      // In ra vi tri chinh phuong
max dau tien
144:      printf("\n Vi tri cp max dau
tien la: %d", In_vtdau_X(a,n,
giatri));
145:      ;
146:      }
147:      // In ra vi tri chinh phuong
cuoi cung
148:
149:      printf("\n Vi tri so max
nguyen to ");
150:      In_vt_X(a,n,max_nt(a,n));
151:      printf("\n Vi tri so max
nguyen to cuoi cung %d",
In_vtcuoi_X(a,n,max_nt(a,n)));
152:
153:      free(a); // giai phong bo nho
154:      return 1;
155:  }
156:
```