

```
1 #include <stdio.h>
2
3 #define SIZE 10
4
5
6 void sort(int arr[], int size)
7 {
8     int i, j, t;
9
10    for (i=0; i<size; i++) {
11        for (j=i+1; j<size; j++) {
12            if (arr[j] < arr[i]) {
13                t = arr[i];
14                arr[i] = arr[j];
15                arr[j] = t;
16            }
17        }
18    }
19 }
20
21
22 float getMedian(int *arr, int size)
23 {
24     if (size % 2 == 0) {
25         return (arr[(size-1)/2] + arr[size/2]) / 2.0;
26     }
27     else {
28         return arr[size/2];
29     }
30 }
31
32
33 int main()
34 {
35     int i;
36     int arr[SIZE];
37
38     //printf("请输入%d个整数: ", SIZE);
39     for (i=0; i<SIZE; i++)
40         scanf("%d", &arr[i]);
41     sort(arr, SIZE);
42     printf("%g", getMedian(arr, SIZE));
43
44     return 0;
45 }
46
```

```
1 #include <stdio.h>
2
3 #define MAX 100
4
5
6 void reverse(int *a, int n)
7 {
8     int i, t;
9
10    for (i=0; i<n/2; i++) {
11        t = a[i];
12        a[i] = a[n-1-i];
13        a[n-1-i] = t;
14    }
15 }
16
17
18 int main()
19 {
20     int a[MAX], n, i;
21
22     scanf("%d", &n);
23     for (i=0; i<n; i++)
24         scanf("%d", a+i);
25     reverse(a, n);
26     for (i=0; i<n; i++)
27         printf("%d ", a[i]);
28
29     return 0;
30 }
31
```

```
1 #include <stdio.h>
2
3 #define MAX 100
4
5
6 int subarr(int *a, int n)
7 {
8     int i, max, maxi, sum;
9
10    maxi = 0;
11    max = a[0] + a[1] + a[2];
12
13    for (i=1; i<n-2; i++) {
14        sum = a[i] + a[i+1] + a[i+2];
15        if (sum > max) {
16            maxi = i;
17            max = sum;
18        }
19    }
20
21    return maxi;
22 }
23
24
25 int main()
26 {
27     int a[MAX], n, i;
28
29     scanf("%d", &n);
30     for (i=0; i<n; i++)
31         scanf("%d", &a[i]);
32     printf("%d", subarr(a, n));
33
34     return 0;
35 }
36
```

```
1 #include <stdio.h>
2
3 #define MAX 50
4
5
6 float InnerProduct(float *a, float *b, int n)
7 {
8     int i;
9     float sum=0;
10
11    for (i=0; i<n; i++)
12        sum += *(a+i) * *(b+i);
13
14    return sum;
15 }
16
17
18 int main()
19 {
20     float a[MAX], b[MAX];
21     int i, n;
22     float product;
23
24     scanf ("%d", &n);
25     for (i=0; i<n; i++)
26         scanf ("%f", a+i);
27     for (i=0; i<n; i++)
28         scanf ("%f", b+i);
29     product = InnerProduct(a, b, n);
30     printf ("%f", product);
31
32     return 0;
33 }
34
```

```
1 #include <stdio.h>
2
3 #define MAX 101
4
5
6 int parentheses(char *s)
7 {
8     int left=0, right=0;
9
10    while (*s) {
11        if (*s == '(')
12            left++;
13        else if (*s == ')')
14            right++;
15        if (left < right)
16            return 0;
17        s++;
18    }
19
20    if (left != right)
21        return 0;
22    return 1;
23 }
24
25
26 int main()
27 {
28     char s[MAX];
29     int i;
30
31     for (i=0; i<MAX-1 && (s[i]=getchar()) != '\n'; i++);
32     s[i] = '\0';
33
34     printf("%s", parentheses(s) ? "true" : "false");
35 }
36
```

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <time.h>
5
6 #define NUM 52
7 #define LEN 5
8 #define SUIT 13
9
10
11 void deal(char *p)
12 {
13     int i, m;
14     char t[LEN];
15
16     srand((unsigned)time(NULL));
17     for (i=0; i<NUM; i++) {
18         m = rand() % (NUM-i) + i;
19         strcpy(t, p+i*LEN);
20         strcpy(p+i*LEN, p+m*LEN);
21         strcpy(p+m*LEN, t);
22     }
23 }
24
25
26 int main()
27 {
28     char poker[NUM][LEN];
29     int i, op;
30
31     for (i=0; i<SUIT; i++) {
32         sprintf(poker[i], "S_%02d", i+1);
33         sprintf(poker[i+SUIT], "H_%02d", i+1);
34         sprintf(poker[i+SUIT*2], "C_%02d", i+1);
35         sprintf(poker[i+SUIT*3], "D_%02d", i+1);
36     }
37
38     while (1) {
39         printf("1-发牌, 0-退出: ");
40         scanf("%d", &op);
41
42         if (op == 1) {
43             deal(*poker);
44             for (i=0; i<NUM; i++)
45                 printf("%s%c", poker[i], (i+1)%SUIT?' ':'\n');
46         }
47         else if (op == 0)
48             return 0;
49     }
50 }
51

```