```c
#include <stdio.h>

#define SIZE 10


void sort(int arr[], int size)
{
    int i, j, t;

    for (i=0; i<size; i++) {
        for (j=i+1; j<size; j++) {
            if (arr[j] < arr[i]) {
                t = arr[i];
                arr[i] = arr[j];
                arr[j] = t;
            }
        }
    }
}


float getMedian(int *arr, int size)
{
    if (size % 2 == 0) {
        return (arr[(size-1)/2] + arr[size/2]) / 2.0;
    }
    else {
        return arr[size/2];
    }
}


int main()
{
    int i;
    int arr[SIZE];

    //printf("请输入%d个整数： ", SIZE);
    for (i=0; i<SIZE; i++)
        scanf("%d", &arr[i]);
    sort(arr, SIZE);
    printf("%g", getMedian(arr, SIZE));

    return 0;
}

```

```c
#include <stdio.h>

#define MAX 100


void reverse(int *a, int n)
{
    int i, t;

    for (i=0; i<n/2; i++) {
        t = a[i];
        a[i] = a[n-1-i];
        a[n-1-i] = t;
    }
}


int main()
{
    int a[MAX], n, i;

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

    return 0;
}

```

```c
#include <stdio.h>

#define MAX 100


int subarr(int *a, int n)
{
    int i, max, maxi, sum;

    maxi = 0;
    max = a[0] + a[1] + a[2];

    for (i=1; i<n-2; i++) {
        sum = a[i] + a[i+1] + a[i+2];
        if (sum > max) {
            maxi = i;
            max  = sum;
        }
    }

    return maxi;
}


int main()
{
    int a[MAX], n, i;

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

    return 0;
}
```

```c
#include <stdio.h>

#define MAX 50


float InnerProduct(float *a, float *b, int n)
{
    int i;
    float sum=0;

    for (i=0; i<n; i++)
        sum += *(a+i) * *(b+i);

    return sum;
}


int main()
{
    float a[MAX], b[MAX];
    int   i, n;
    float product;

    scanf("%d", &n);
    for (i=0; i<n; i++)
        scanf("%f", a+i);
    for (i=0; i<n; i++)
        scanf("%f", b+i);
    product = InnerProduct(a, b, n);
    printf("%f", product);

    return 0;
}

```

```c
#include <stdio.h>

#define MAX 101


int parentheses(char *s)
{
    int left=0, right=0;

    while (*s) {
        if (*s == '(')
            left++;
        else if (*s == ')')
            right++;
        if (left < right)
            return 0;
        s++;
    }

    if (left != right)
        return 0;
    return 1;
}


int main()
{
    char s[MAX];
    int  i;

    for (i=0; i<MAX-1 && (s[i]=getchar())!='\n'; i++);
    s[i] = '\0';

    printf("%s", parentheses(s) ? "true" : "false");
}

```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define NUM  52
#define LEN   5
#define SUIT 13


void deal(char *p)
{
    int i, m;
    char t[LEN];

    srand((unsigned)time(NULL));
    for (i=0; i<NUM; i++) {
        m = rand() % (NUM-i) + i;
        strcpy(      t, p+i*LEN);
        strcpy(p+i*LEN, p+m*LEN);
        strcpy(p+m*LEN,       t);
    }
}


int main()
{
    char poker[NUM][LEN];
    int  i, op;

    for (i=0; i<SUIT; i++) {
        sprintf(poker[i       ], "S_%02d", i+1);
        sprintf(poker[i+SUIT  ], "H_%02d", i+1);
        sprintf(poker[i+SUIT*2], "C_%02d", i+1);
        sprintf(poker[i+SUIT*3], "D_%02d", i+1);
    }

    while (1) {
        printf("1-发牌, 0-退出: ");
        scanf("%d", &op);

        if (op == 1) {
            deal(*poker);
            for (i=0; i<NUM; i++)
                printf("%s%c", poker[i], (i+1)%SUIT?' ':'\n');
        }
        else if (op == 0)
            return 0;
    }
}

```