

// 计算两个整数的最大公约数

#include <stdio.h>

int max\_common\_factor(int m, int n);

int main()

{

  int m = 12, n = 18;

  printf("The max common factor of %d and %d is %d\n", m, n, max\_common\_factor(m, n));

  return 0;

}

int max\_common\_factor(int m, int n)

{

  int temp;

  while (n != 0)

  {

    temp = m % n;

    m = n;

    n = temp;

  }

  return m;

}

// 定义函数GetData() 用于接收键盘输入的一组整型数据, 并放入一个数组中;

// 另外再定义一个函数Sort() 用于对输入的这一组数据进行降序排列。

// 主函数先后调用GetData和 Sort函数, 输出最后的排序结果。

#include <stdio.h>

int GetData(int \*arr, int n);

void Sort(int \*arr, int n);

int main()

{

  int n = 5;

  int arr[n];

  GetData(arr, n);

  Sort(arr, n);

  for (int i = 0; i < n; i++)

  {

    printf("%d ", arr[i]);

  }

  printf("\n");

  return 0;

}

int GetData(int \*arr, int n)

{

  for (int i = 0; i < n; i++)

  {

    printf("Please input an integer: ");

    scanf("%d", &arr[i]);

  }

  return 0;

}

// 冒泡排序

void Sort(int \*arr, int n)

{

  int temp;

  for (int i = 0; i < n - 1; i++)

  {

    for (int j = 0; j < n - 1 - i; j++)

    {

      if (arr[j] < arr[j + 1])

      {

        temp = arr[j];

        arr[j] = arr[j + 1];

        arr[j + 1] = temp;

      }

    }

  }

}

// 编制函数JsSort(), 其函数的功能是 :

// 对字符串变量的下标为奇数的字符按其ASCII值从大到小的顺序进行排序, 排序后的结果仍存入字符串数组中。

// 例如 : 位置 01234567 源字符串 abcdefgh 处理后字符串ahcfedgb

#include <stdio.h>

#include <string.h>

void JsSort(char \*str);

int main()

{

  char str[] = "abcdefgh";

  JsSort(str);

  printf("%s\n", str);

  return 0;

}

void JsSort(char \*str)

{

  char temp;

  int length = strlen(str);

  for (int i = 1; i < length; i += 2)

  {

    for (int j = i + 2; j < length; j += 2)

    {

      if (str[i] < str[j])

      {

        temp = str[i];

        str[i] = str[j];

        str[j] = temp;

      }

    }

  }

}

// 编写一函数StrLoc, 其功能是求得一字符串strl在另一字符串str2中首次出现的位置,

// 如果 str1 不在 str2 中,则返回一1。例如,假设 strl为"do",str2 为"how do you do?",则返回值为4

#include <stdio.h>

int StrLoc(const char \*str1, const char \*str2);

int main()

{

  char str1[] = "do";

  char str2[] = "how do you do?";

  printf("The position of %s in %s is %d\n", str1, str2, StrLoc(str1, str2));

  return 0;

}

int StrLoc(const char \*str1, const char \*str2)

{

  int i, j;

  for (i = 0; str2[i] != '\0'; i++)

  {

    for (j = 0; str1[j] != '\0'; j++)

    {

      if (str1[j] != str2[i + j])

      {

        break;

      }

    }

    if (str1[j] == '\0')

    {

      return i + 1;

    }

  }

  return -1;

}

// 定义函数GetData() 用于接收键盘输入的一组整型数据, 并放入一个数组中;

// 另外再定义一个函数Sort() 用于对输入的这一组数据进行降序排列。

// 主函数先后调用GetData和 Sort函数, 输出最后的排序结果。

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

// 随机产生n个整数,存入数组arr中

int GetData(int \*arr, int n);

void Sort(int \*arr, int n);

int main()

{

  int n = 5;

  int arr[n];

  GetData(arr, n);

  Sort(arr, n);

  for (int i = 0; i < n; i++)

  {

    printf("%d ", arr[i]);

  }

  printf("\n");

  return 0;

}

int GetData(int \*arr, int n)

{

  for (int i = 0; i < n; i++)

  {

    // printf("Please input an integer: ");

    // scanf("%d", &arr[i]);

    arr[i] = rand() % 100;

  }

  return 0;

}

// 冒泡排序

void Sort(int \*arr, int n)

{

  int temp;

  for (int i = 0; i < n - 1; i++)

  {

    for (int j = 0; j < n - 1 - i; j++)

    {

      if (arr[j] < arr[j + 1])

      {

        temp = arr[j];

        arr[j] = arr[j + 1];

        arr[j + 1] = temp;

      }

    }

  }

}

// 编写函数 fun(int x, int y, intz, intn),

// 其功能是从x个红球、y个白球、z个黑球中任意取出n个, 且其中必须要有红球和白球, 要求输出所有方案。

#include <stdio.h>

void fun(int x, int y, int z, int n);

int main()

{

  fun(2, 3, 4, 3);

  return 0;

}

void fun(int x, int y, int z, int n)

{

  int i, j, k;

  for (i = 1; i <= x; i++) // 至少一个红球

  {

    for (j = 1; j <= y; j++) // 至少一个白球

    {

      for (k = 0; k <= z; k++) // 可以没有黑球

      {

        if (i + j + k == n)

        {

          printf("Red: %d, White: %d, Black: %d\n", i, j, k);

        }

      }

    }

  }

}