

第二次作业

author: Hommer

date: 0927

1.输入4个整数，要求按由小到大的顺序输出。（可采用任意算法，但建议采用数组及两重循环方法）

```
#include <stdio.h>

int main() {
    int a[4], i, j, min_idx, tmp;
    printf("Please enter 4 integers (separated by spaces or newlines): ");
    scanf("%d %d %d %d", &a[0], &a[1], &a[2], &a[3]);

    // Simple selection sort: ascending order
    for (i = 0; i < 3; ++i) {
        min_idx = i;
        for (j = i + 1; j < 4; ++j) {
            if (a[j] < a[min_idx]) min_idx = j;
        }
        if (min_idx != i) {
            tmp = a[i];
            a[i] = a[min_idx];
            a[min_idx] = tmp;
        }
    }

    printf("Sorted ascending: %d %d %d %d\n", a[0], a[1], a[2], a[3]);
    return 0;
}
```

2. 编程，输入一个十进制正整数，将其转换为二进制整数输出。（输入数据范围在0-100之间）

```
#include <stdio.h>

int main() {
    int n, i;
    printf("Please enter a decimal integer (0-100): ");
    scanf("%d", &n);

    if (n < 0 || n > 100) {
        printf("Number out of range (0-100).\n");
        return 1;
    }

    if (n == 0) {
        printf("Binary: 0\n");
        return 0;
    }

    int bits[7]; // 100 < 128, so 7 bits suffice
    int cnt = 0;
    while (n > 0) {
        bits[cnt++] = n % 2;
        n = n / 2;
    }

    printf("Binary: ");
    for (i = cnt - 1; i >= 0; --i) {
        putchar(bits[i] ? '1' : '0');
    }
    putchar('\n');

    return 0;
}
```

3. 根据求 π 近似值的公式，计算并输出 π 的值，设 $n=1000$ 。

$$\pi = 2 \times \dots \times \frac{(2n)^2}{(2n+1)(2n-1)}$$

```
#include <stdio.h>

int main() {
    const int n = 1000;
    int k;
    double prod = 1.0, num, den;

    for (k = 1; k <= n; ++k) {
        num = 4.0 * k * k;
        den = (2.0 * k - 1.0) * (2.0 * k + 1.0);
        prod *= num / den;
    }

    double pi = 2.0 * prod;
    printf("n = %d, pi = %.18f\n", n, pi);
    return 0;
}
```

$n = 1000$, $\pi = 3.140807746030402000$

4. 打印出所有的 "水仙花数"，所谓 "水仙花数"是指一个三位数，其各位数字立方和等于该数本身。例如：153是一

个 "水仙花数 "，因为 $153=1^3+5^3+3^3$ 。

```
#include <stdio.h>

int main() {
    int n, a, b, c;
    printf("Armstrong numbers:\n");
    for (n = 100; n <= 999; ++n) {
        a = n / 100;
        b = (n / 10) % 10;
        c = n % 10;
        if (a*a*a + b*b*b + c*c*c == n) {
            printf("%d\n", n);
        }
    }
    return 0;
}
```

Armstrong numbers:

153
370
371
407

5. 一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如，6的因子为1，2，3，而 $6=1+2+3$ ，因此6是“完数”。

数”。编程，找出1000之内的所有完数，并按以下格式输出其因子：完数6的因子为：1，2，3。

```
#include <stdio.h>

int main() {
    int n, i, j;
    printf("Perfect numbers (<=1000) and their divisors:\n");
    for (n = 2; n < 1000; ++n) {
        int sum = 0;
        int divisors[32];
        int cnt = 0;
        for (i = 1; i <= n / 2; ++i) {
            if (n % i == 0) {
                sum += i;
                if (cnt < 32) divisors[cnt++] = i;
            }
        }
        if (sum == n) {
            printf("Perfect number %d has divisors: ", n);
            for (j = 0; j < cnt; ++j) {
                if (j) printf(", ");
                printf("%d", divisors[j]);
            }
            printf(".\n");
        }
    }
    return 0;
}
```

Perfect numbers (<=1000) and their divisors:

Perfect number 6 has divisors: 1, 2, 3.

Perfect number 28 has divisors: 1, 2, 4, 7, 14.

Perfect number 496 has divisors: 1, 2, 4, 8, 16, 31, 62, 124, 248.

6.从键盘输入10个学生的学号和三门课成绩，计算每个学生的平均成绩并输出其学号和平均成绩，同时找出其中平

均成绩最高和最低的学生并输出其学号和平均成绩。自定义数据结构。（思考不同数据结构的优缺点）

```
#include <stdio.h>

#define N_STUDENTS 10

int main(void) {
    int ids[N_STUDENTS];
    double scores[N_STUDENTS][3];
    double avg[N_STUDENTS];
    int i;

    printf("Please enter %d students' ID and three scores (ID s1 s2 s3) one per line:\n", N_STU
    for (i = 0; i < N_STUDENTS; ++i) {
        printf("Student %d: ", i + 1);
        if (scanf("%d %lf %lf %lf", &ids[i],
                    &scores[i][0], &scores[i][1], &scores[i][2]) != 4) {
            printf("Invalid input. Exiting.\n");
            return 1;
        }
        avg[i] = (scores[i][0] + scores[i][1] + scores[i][2]) / 3.0;
    }

    int max_idx = 0, min_idx = 0;
    for (int i = 1; i < N_STUDENTS; ++i) {
        if (avg[i] > avg[max_idx]) max_idx = i;
        if (avg[i] < avg[min_idx]) min_idx = i;
    }

    printf("\nID\tAverage\n");
    for (i = 0; i < N_STUDENTS; ++i) {
        printf("%d\t%.2f\n", ids[i], avg[i]);
    }

    printf("\nHighest average: ID %d, Average %.2f\n", ids[max_idx], avg[max_idx]);
    printf("Lowest average: ID %d, Average %.2f\n", ids[min_idx], avg[min_idx]);

    return 0;
}
```

7.下面程序的作用是计算两个三维矢量的内积。两个矢量分别存放在两个数组中，从键盘读入两个矢量，并用for循环完成内积的计算，请补全程序中空缺的部分。

```
#include <stdio.h>

int main() {
    float a[3], b[3], dotProd = 0; int i;
    printf("请输入第一个矢量: "); scanf(_____);
    printf("请输入第二个矢量: "); scanf(_____);
    for (_____) {
        _____; }
    printf("内积是: %f\n", dotProd);
}
```

```
#include <stdio.h>

int main() {
    float a[3], b[3], dotProd = 0; int i;
    printf("请输入第一个矢量: ");
    scanf("%f %f %f", &a[0], &a[1], &a[2]);
    printf("请输入第二个矢量: ");
    scanf("%f %f %f", &b[0], &b[1], &b[2]);
    for (i = 0; i < 3; ++i) {
        dotProd += a[i] * b[i];
    }
    printf("内积是: %f\n", dotProd);
    return 0;
}
```