

编程题（第一题 10 分，第二题 14 分，共 24 分）

1. 编写程序，将一个正整数分解质因数。例如：输入 90, 打印出 $90=2*3*3*5$ 。

提示：对 n 进行分解质因数，应先找到一个最小的质数 k ，然后按下述步骤完成：

(1) 如果这个质数恰等于 n ，则说明分解质因数的过程已经结束，打印出即可。

(2) 如果 $n > k$ ，但 n 能被 k 整除，则应打印出 k 的值，并用 n 除以 k 的商,作为新的正整数你 n ,重复执行第一步。

(3) 如果 n 不能被 k 整除，则用 $k+1$ 作为 k 的值,重复执行第一步。

参考答案

```
#include <stdio.h>

//因式分解函数 8 分，以下三个 fact 函数及评分细则仅做参考
void fact1(int n, int k) //2 分
{
    if (n == 1) {
        printf("1"); // 考虑特殊情况
        return;
    }
    if (k == n) printf("%d", k); //2 分
    else {
        if (n % k == 0) //2 分
        {
            printf("%d", k);
            printf("*");
            fact1(n / k, k);
        }
        else //2 分
        {
            fact1(n, k + 1);
        }
    }
}

void fact2(int n) { //2 分
    if (n == 1) {
```

```

        printf("1"); // 考虑特殊情况
        return;
    }
    int i;
    for (i = 2; i <= n; i++) { //2 分
        if (n % i == 0) break;
    }
    printf("%d", i); //1 分
    if (i < n) { //1 分
        printf("*"); //1 分
        fact2(n / i); //1 分
    }
}

void fact3(int n) { //2 分
    if (n == 1) {
        printf("1"); // 考虑特殊情况
        return;
    }
    int i = 2;
    do {
        for (; i <= n; i++) //2 分
            if (n % i == 0) break;
        printf("%d", i); //1 分
        if (i < n) { //2 分
            printf("*");
            n = n / i;
        }
        else { //1 分
            break;
        }
    } while (1);
}

```

//main 函数 2 分

int main() { //1 分

int n;

printf("please input n:");

scanf_s("%d", &n);

printf("%d=", n);

printf("fact1:\n");

fact1(n, 2); //1 分

printf("\nfact2:\n");

fact2(n);

printf("\nfact3:\n");

fact3(n);

return 0;

}

2. 不用工作数组，分别编写实现将已知方阵旋转 90° 的函数。

参考答案

```
#include <stdio.h>
```

```
#define N 8
```

```
int a[N][N];
```

```
void main(){
```

```
int i, j, k = 1, t, n = N;
```

```
printf("这里以 8*8 的方阵为例\n");
```

```
for (i = 0; i < N; i++)
```

```
    for (j = 0; j < N; j++)
```

```
        a[i][j] = k++;
```

```
printf("顺时针转之前的矩阵如下: \n");
```

```
for (i = 0; i < N; i++)
```

```
{
```

```
    for (j = 0; j < N; j++)
```

```
        printf("%4d", a[i][j]);
```

```
    printf("\n");
```

```
}
```

```

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

```

printf("顺时针转之后的矩阵如下: \n");

```

for (i = 0; i < N; i++){
    for (j = 0; j < N; j++)
        printf("%4d", a[i][j]);
    printf("\n");
}

```

printf("\n");

```

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

```

printf("继续逆时针转之后的矩阵如下: \n");

```

for (i = 0; i < N; i++){
    for (j = 0; j < N; j++)
        printf("%4d", a[i][j]);
    printf("\n");
}

```

printf("\n");

}