4.5

// Created by 刘家琪 on 15/3/22.

#include <stdio.h>

int N = 4;

int P[10];

int main()

{

void Perm1(int M);

int i;

//对数组初始化

for(i = 1; i <= N; i++)

P[i] = i;

//调用排序函数

Perm1(1);

return 0;

}

排列函数

void Perm1(int M)

{

void Print(int P[]);

int i,j,k;

if(M == N)

Print(P);

else

{

for(i = M; i <= N; i++)

{

//将第i个元素放置还未排列的数组的首位，第i个元素前面的元素依次后移，以便按字典序输出

k = P[i];

for(j = i; j > M; j--)

P[j] = P[j-1];

P[j] = k;

//递归调用

Perm1(M+1);

//将数组已交换的元素还原到原位置，以免发生混乱

k = P[M];

for(j = M; j < i; j++)

P[j] = P[j+1];

P[j] = k;

}

}

}

//输出函数

void Print(int P[])

{

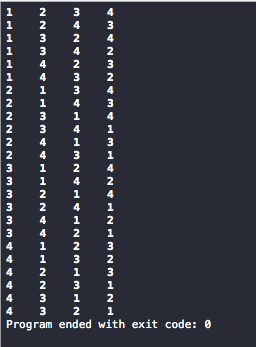
int i;

for(i = 1; i <= N;i++)

printf("%d\t",P[i]);

printf("\n");

}

输出结果展示：

4.7

// Created by 刘家琪 on 15/3/22.

#include <stdio.h>

int N = 4;

int P[10];

int main()

{

void Perm2(int M);

int i;

//对数组初始化

for(i = 1; i <= N; i++)

{

P[i] = 0;

}

//调用排序函数

Perm2(1);

return 0;

}

void Perm2(int M)

{

void Print(int P[]);

int i;

if(M == N+1)

Print(P);

else

{

for(i = 1; i <= N; i++)

{

if(P[i] == 0)

{

P[i] = M;

Perm2(M+1);

P[i] = 0;

}

}

}

}

void Print(int P[])

{

int i;

for(i = 1; i <= N;i++)

printf("%d\t",P[i]);

printf("\n");

}

输出结果展示：

