#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define Maxsize 100

typedef struct parts {

    int partnumber;

    char partname[Maxsize];

    int partquantity;

}Parts;

int num\_parts = 0,count=1;

void InitParts(Parts parts[]);

void InsertParts(Parts parts[]);

void SearchParts(Parts parts[]);

void UpdataParts(Parts parts[]);

void DeleteParts(Parts parts[]);

void PrintParts(Parts parts[]);

int main()

{

    Parts\* parts = (Parts\*)malloc(sizeof(Parts) \* 10);

    InitParts(parts);

    char code = 0;

    do

    {

        printf("Enter operation code:");

        scanf(" %c", &code);

        switch (code)

        {

        case 'i':InsertParts(parts); break;

        case 's':SearchParts(parts); break;

        case 'u':UpdataParts(parts); break;

        case 'p':PrintParts(parts); break;

        case 'd':DeleteParts(parts); break;

        case 'q':break;

        default:break;

        }

        if (num\_parts > 10)

        {

            count++;

            parts = (Parts\*)realloc(parts,sizeof(Parts) \* 10\*count);

            InitParts(parts);

        }

    } while (code != 'q');

    free(parts);

    return 0;

}

void InitParts(Parts parts[])

{

    int i = 0;

    for (i = (count-1)\*10; i< count\*10; i++)

    {

        parts[i].partnumber = 0;

        parts[i].partquantity = 0;

        char partname[Maxsize] = { '\0' };

        strcpy(&parts[i].partname, partname);

    }

}

void InsertParts(Parts parts[])

{

    printf("Enter part number:");

    scanf("%d", &parts[num\_parts].partnumber);

    printf("Enter part name:");

    getchar();

    gets(&parts[num\_parts].partname);

    printf("Enter quantity on hand:");

    scanf("%d", &parts[num\_parts].partquantity);

    num\_parts++;

}

void SearchParts(Parts parts[])

{

    int num = 0,i=0;

    printf("Enter part number:");

    scanf("%d", &num);

    while (parts[i].partnumber!=num&& parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[i].partnumber != 0)

    {

        printf("Part name :%s\n", parts[i].partname);

        printf("Quantity on hand:%d\n", parts[i].partquantity);

    }

    else

    {

        printf("Part not found.\n");

    }

}

void UpdataParts(Parts parts[])

{

    int num = 0,q=0,i=0;

    printf("Enter part number:");

    scanf("%d", &num);

    while (parts[i].partnumber != num&& parts[i].partnumber!=0)

    {

        i++;

    }

    if (parts[i].partnumber != 0)

    {

        printf("Enter change in quantity on hand:");

        scanf("%d", &q);

        parts[i].partquantity += q;

    }

    else

    {

        printf("Error! It is not in the database!\n");

    }

}

void PrintParts(Parts parts[])

{

    printf("Part Number Part Name Quantity on Hand\n");

    int i = 0;

    while (parts[i].partnumber != 0)

    {

        printf("%d           %s             %d\n", parts[i].partnumber, parts[i].partname, parts[i].partquantity);

        i++;

    }

}

void DeleteParts(Parts parts[])

{

    int num = 0, i = 0;

    printf("Enter the part number you want to delete:");

    scanf("%d", &num);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[0].partnumber == 0)

    {

        printf("Error!The database is empty!\n");

    }

    else if(parts[i].partnumber == 0)

    {

        printf("Error!The part is not in the database!\n");

    }

    else

    {

        while (parts[++i].partnumber == 0)

        {

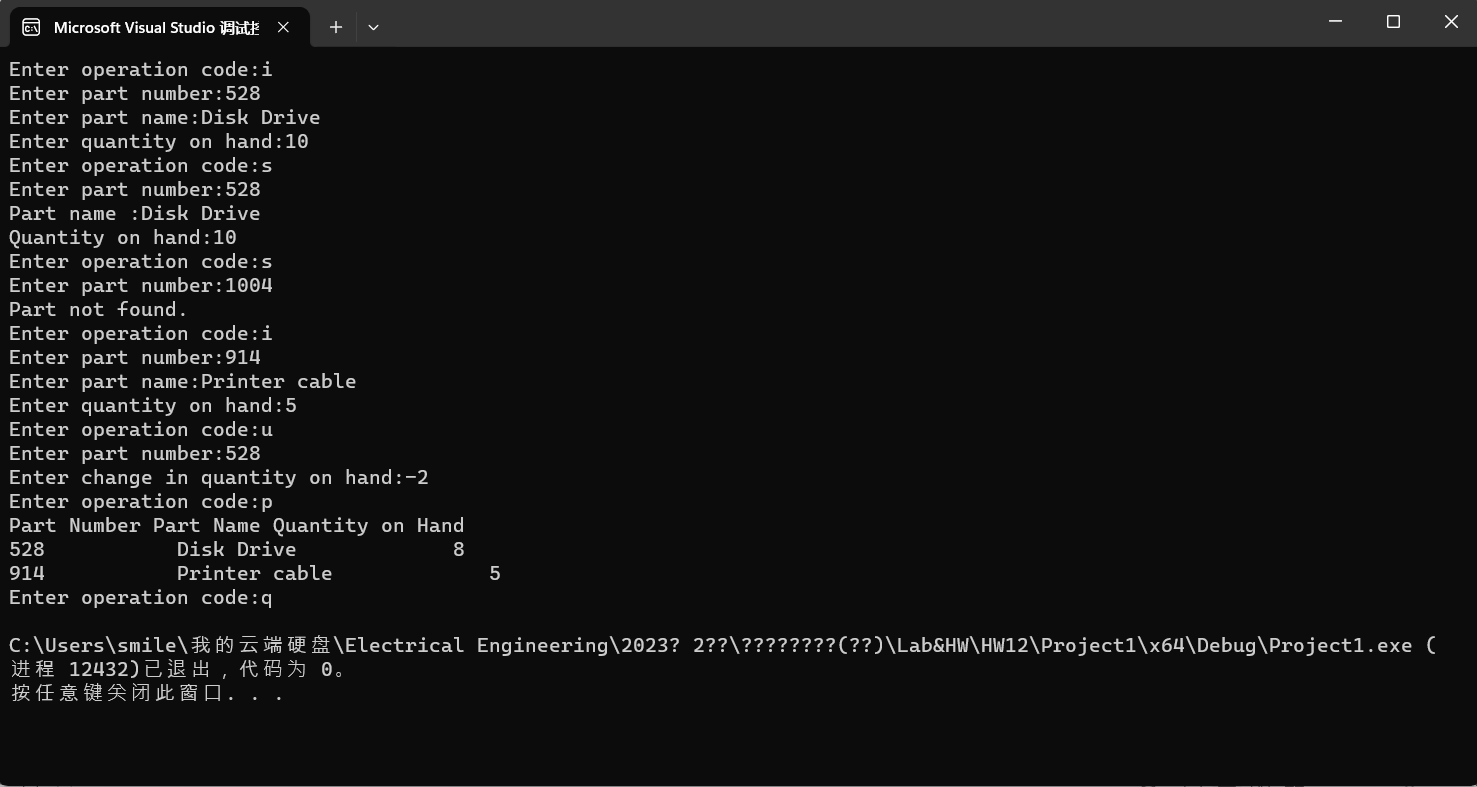
            parts[i-    1] = parts[i];

        }

        num\_parts--;

    }

}



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define Maxsize 100

typedef struct parts {

    int partnumber;

    char partname[Maxsize];

    int partquantity;

    char category[25];

}Parts;

int num\_parts = 0, count = 1;

void InitParts(Parts parts[]);

void InsertParts(Parts parts[]);

void SearchParts(Parts parts[]);

void UpdataParts(Parts parts[]);

void DeleteParts(Parts parts[]);

void PrintParts(Parts parts[]);

int main()

{

    Parts\* parts = (Parts\*)malloc(sizeof(Parts) \* 10);

    InitParts(parts);

    char code = 0;

    do

    {

        printf("Enter operation code:");

        scanf(" %c", &code);

        switch (code)

        {

        case 'i':InsertParts(parts); break;

        case 's':SearchParts(parts); break;

        case 'u':UpdataParts(parts); break;

        case 'p':PrintParts(parts); break;

        case 'd':DeleteParts(parts); break;

        case 'q':break;

        default:break;

        }

        if (num\_parts > 10)

        {

            count++;

            parts = (Parts\*)realloc(parts, sizeof(Parts) \* 10 \* count);

            InitParts(parts);

        }

    } while (code != 'q');

    free(parts);

    return 0;

}

void InitParts(Parts parts[])

{

    int i = 0;

    for (i = (count - 1) \* 10; i < count \* 10; i++)

    {

        parts[i].partnumber = 0;

        parts[i].partquantity = 0;

        char partname[Maxsize] = { '\0' };

        strcpy(&parts[i].partname, partname);

        strcpy(&parts[i].category, partname);

    }

}

void InsertParts(Parts parts[])

{

    printf("Enter part number:");

    scanf("%d", &parts[num\_parts].partnumber);

    printf("Enter part name:");

    getchar();

    gets(&parts[num\_parts].partname);

    printf("Enter category:");

    gets(&parts[num\_parts].category);

    printf("Enter quantity on hand:");

    scanf("%d", &parts[num\_parts].partquantity);

    getchar();

    num\_parts++;

}

void SearchParts(Parts parts[])

{

    int num = 0, i = 0;

    printf("Enter part number:");

    scanf("%d", &num);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[i].partnumber != 0)

    {

        printf("Part name :%s\n", parts[i].partname);

        printf("Category name :%s\n", parts[i].category);

        printf("Quantity on hand:%d\n", parts[i].partquantity);

    }

    else

    {

        printf("Part not found.\n");

    }

}

void UpdataParts(Parts parts[])

{

    int num = 0, q = 0, i = 0;

    printf("Enter part number:");

    scanf("%d", &num);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[i].partnumber != 0)

    {

        printf("Enter change in quantity on hand:");

        scanf("%d", &q);

        parts[i].partquantity += q;

    }

    else

    {

        printf("Error! It is not in the database!\n");

    }

}

void PrintParts(Parts parts[])

{

    printf("Part Number    Part Name Category name Quantity on Hand \n");

    int i = 0;

    while (parts[i].partnumber != 0)

    {

        printf("%d           %s             %s             %d\n", parts[i].partnumber, parts[i].partname, parts[i].category,parts[i].partquantity);

        i++;

    }

}

void DeleteParts(Parts parts[])

{

    int num = 0, i = 0;

    printf("Enter the part number you want to delete:");

    scanf("%d", &num);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[0].partnumber == 0)

    {

        printf("Error!The database is empty!\n");

    }

    else if (parts[i].partnumber == 0)

    {

        printf("Error!The part is not in the database!\n");

    }

    else

    {

        while (parts[++i].partnumber == 0)

        {

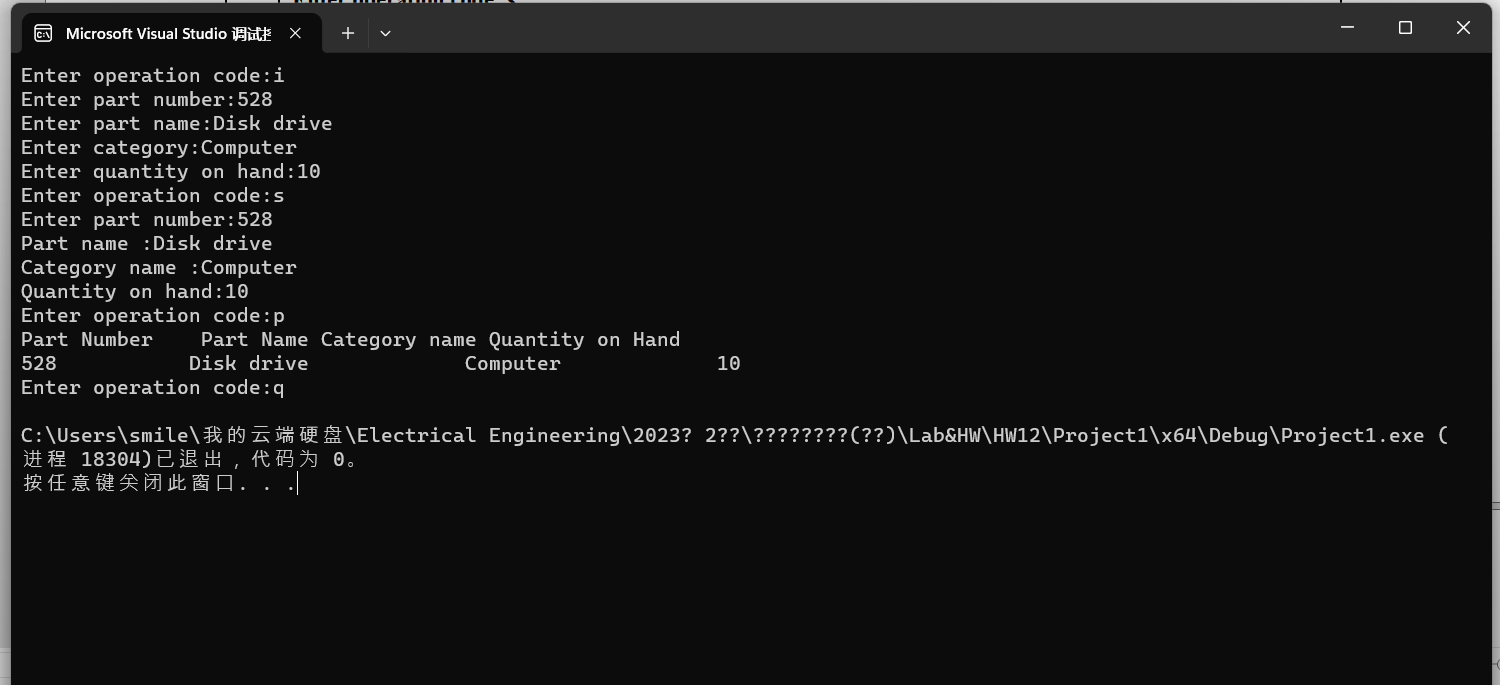
            parts[i - 1] = parts[i];

        }

        num\_parts--;

    }

}



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define Maxsize 100

typedef struct parts {

    int partnumber;

    char partname[Maxsize];

    int partquantity;

    char category[25];

}Parts;

int num\_parts = 0, count = 1;

void InitParts(Parts parts[]);

void InsertParts(Parts parts[]);

void SearchParts(Parts parts[]);

void UpdataParts(Parts parts[]);

void DeleteParts(Parts parts[]);

void ChangeParts(Parts parts[]);

void PrintParts(Parts parts[]);

int main()

{

    Parts\* parts = (Parts\*)malloc(sizeof(Parts) \* 10);

    InitParts(parts);

    char code = 0;

    do

    {

        printf("Enter operation code:");

        scanf(" %c", &code);

        switch (code)

        {

        case 'i':InsertParts(parts); break;

        case 's':SearchParts(parts); break;

        case 'u':UpdataParts(parts); break;

        case 'p':PrintParts(parts); break;

        case 'd':DeleteParts(parts); break;

        case 'c':ChangeParts(parts); break;

        case 'q':break;

        default:break;

        }

        if (num\_parts > 10)

        {

            count++;

            parts = (Parts\*)realloc(parts, sizeof(Parts) \* 10 \* count);

            InitParts(parts);

        }

    } while (code != 'q');

    free(parts);

    return 0;

}

void InitParts(Parts parts[])

{

    int i = 0;

    for (i = (count - 1) \* 10; i < count \* 10; i++)

    {

        parts[i].partnumber = 0;

        parts[i].partquantity = 0;

        char partname[Maxsize] = { '\0' };

        strcpy(&parts[i].partname, partname);

        strcpy(&parts[i].category, partname);

    }

}

void InsertParts(Parts parts[])

{

    printf("Enter part number:");

    scanf("%d", &parts[num\_parts].partnumber);

    printf("Enter part name:");

    getchar();

    gets(&parts[num\_parts].partname);

    printf("Enter category:");

    gets(&parts[num\_parts].category);

    printf("Enter quantity on hand:");

    scanf("%d", &parts[num\_parts].partquantity);

    getchar();

    num\_parts++;

}

void SearchParts(Parts parts[])

{

    int num = 0, i = 0;

    printf("Enter part number:");

    scanf("%d", &num);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[i].partnumber != 0)

    {

        printf("Part name :%s\n", parts[i].partname);

        printf("Category name :%s\n", parts[i].category);

        printf("Quantity on hand:%d\n", parts[i].partquantity);

    }

    else

    {

        printf("Part not found.\n");

    }

}

void UpdataParts(Parts parts[])

{

    int num = 0, q = 0, i = 0;

    printf("Enter part number:");

    scanf("%d", &num);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[i].partnumber != 0)

    {

        printf("Enter change in quantity on hand:");

        scanf("%d", &q);

        parts[i].partquantity += q;

    }

    else

    {

        printf("Error! It is not in the database!\n");

    }

}

void PrintParts(Parts parts[])

{

    printf("Part Number    Part Name Category name Quantity on Hand \n");

    int i = 0;

    while (parts[i].partnumber != 0)

    {

        printf("%d           %s             %s             %d\n", parts[i].partnumber, parts[i].partname, parts[i].category, parts[i].partquantity);

        i++;

    }

}

void DeleteParts(Parts parts[])

{

    int num = 0, i = 0;

    printf("Enter the part number you want to delete:");

    scanf("%d", &num);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[0].partnumber == 0)

    {

        printf("Error!The database is empty!\n");

    }

    else if (parts[i].partnumber == 0)

    {

        printf("Error!The part is not in the database!\n");

    }

    else

    {

        while (parts[++i].partnumber == 0)

        {

            parts[i - 1] = parts[i];

        }

        num\_parts--;

    }

}

void ChangeParts(Parts parts[])

{

    int num = 0, i = 0;

    printf("Enter part number:");

    scanf("%d", &num);

    printf("Enter the new Part Name:");

    getchar();

    char newpartname[Maxsize];

    gets(newpartname);

    while (parts[i].partnumber != num && parts[i].partnumber != 0)

    {

        i++;

    }

    if (parts[i].partnumber != 0)

    {

        strcpy(&parts[i].partname, newpartname);

    }

    parts = (Parts\*)realloc(parts, sizeof(Parts) \* 10 \* count);

}

