



```
short 2
int 4
long 4
long long 8
unsigned short 2
unsigned int 4
unsigned long 4
char 1
Bool 1
double 8
long double 16
float 4
```

[illegible]

## 第二题

源代码:

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

/*主函数*/

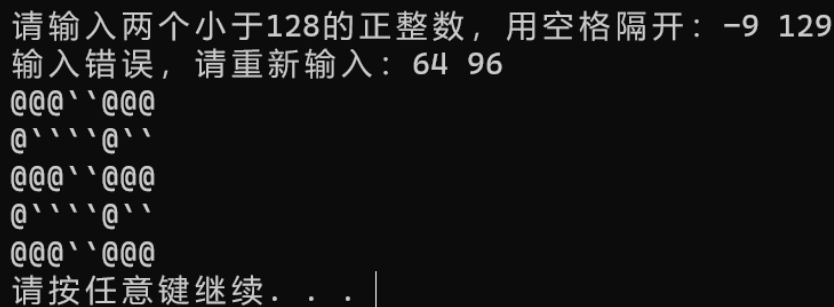
int main() {
    char a=64, b=94; /*定义符号变量*/
    int c, d; /*定义暂存保护变量*/
    printf("请输入两个小于128的正整数，用空格隔开：");
    scanf("%d%d", &c, &d);
    /*若输入的数不正确，进行重新输入*/
    while (c<1 || c>127 || d<1 || d>127) {
        printf("输入错误，请重新输入：");
        scanf("%d%d", &c, &d);
    }
    /*将输入值赋给符号变量*/
    a=(char)c;
    b=(char)d;
    /*输出图案*/
    printf("%c%c%c%c%c%c%c%c\n", a, a, a, b, b, a, a, a);
```

```

printf("%c%c%c%c%c%c%c%c\n", a, b, b, b, b, a, b, b);
printf("%c%c%c%c%c%c%c%c\n", a, a, a, b, b, a, a, a);
printf("%c%c%c%c%c%c%c%c\n", a, b, b, b, b, a, b, b);
printf("%c%c%c%c%c%c%c%c\n", a, a, a, b, b, a, a, a);
system("pause");/*暂停*/
return 0;
}

```

执行结果截图：



```

请输入两个小于128的正整数，用空格隔开：-9 129
输入错误，请重新输入：64 96
@@@`@@@
@`@@@
@@@`@@@
@`@@@
@@@`@@@
@`@@@
@@@`@@@
请按任意键继续. . . |

```

### 第三题

源代码：

```

#include<stdlib.h>
#include<stdio.h>
int main() {
    char cMonth;//月份序号
    unsigned short usSheepTotal;//总羊数
    unsigned short usSheepAdultPair=0;//发育成熟的羊的对数
    unsigned short usSheepAdolescentPair=1;//未发育成熟的羊的对数
    int iMoney;//在当月卖掉所有羊可以换取的钱数
    for (cMonth=1; cMonth<=6; ++cMonth) {
        usSheepTotal=2*(usSheepAdolescentPair+usSheepAdultPair);//当月总羊数的
        计算
        iMoney=10*usSheepTotal;//当月换算钱数的计算
        printf("%u月：羊共%u只，可以卖%d元。\\n", cMonth, usSheepTotal, iMoney);//
        输出当月情况
        usSheepAdultPair+=usSheepAdolescentPair;//下个月小羊发育成熟
        usSheepAdolescentPair=usSheepAdultPair;//下个月每对发育成熟的羊都能生一
        对小羊
    }
    iMoney>=1E5?printf("太棒了，6个月可以筹齐税费。"):printf("太惨了，6个月不能
    筹齐税费。");//将钱数与十万比较，判断能否筹齐税费
    system("pause");
    return 0;
}

```

执行结果截图：

```
1月：羊共2只，可以卖20元。
2月：羊共4只，可以卖40元。
3月：羊共8只，可以卖80元。
4月：羊共16只，可以卖160元。
5月：羊共32只，可以卖320元。
6月：羊共64只，可以卖640元。
太惨了，6个月不能筹齐税费。请按任意键继续...
```

### 选做题

方案 1: 使用 long long 数据类型

源代码:

```
#include<stdlib.h>
#include<stdio.h>
int main() {
    long long llPhone=0;
    printf("请输入十一位手机号码: ");
    scanf("%lld",&llPhone);
    //如果输入的不是十一位:
    while (llPhone<1E10||llPhone>=1E11) {
        printf("输入错误, 请重新输入: ");
        scanf("%lld",&llPhone);
    }
    printf("输入的手机号码: %lld\n",llPhone);
    system("pause");
    return 0;
}
```

执行结果截图:

```
请输入十一位手机号码: 123456789
输入错误, 请重新输入: 83882381314
输入的手机号码: 83882381314
请按任意键继续. . . |
```

方案 2: 使用 unsigned short 数据类型, 分三块存储

源代码:

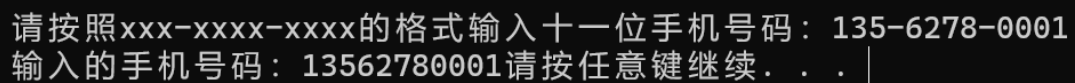
```
#include<stdlib.h>
#include<stdio.h>
int main() {
    unsigned short usST,usND,usRD;
    printf("请按照xxx-xxxx-xxxx的格式输入十一位手机号码: ");
    scanf("%u-%u-%u",&usST,&usND,&usRD);
    //如果输入的格式不正确:
    while (usST<1E2||usST>=1E3||usND>=1E4||usRD>=1E4) {
```

```

    printf("输入错误，请重新输入：");
    scanf("%u-%u-%u", &usST, &usND, &usRD);
}
printf("输入的手机号码： %u", usST); //输出前三位
//输出中间四位
if (usND>=1E3) printf("%u", usND);
    else if (usND>=1E2) printf("0%u", usND);
        else if (usND>=1E1) printf("00%u", usND);
            else if (usND>=1) printf("000%u", usND);
                else printf("0000");
//输出后四位
if (usRD>=1E3) printf("%u", usRD);
    else if (usRD>=1E2) printf("0%u", usRD);
        else if (usRD>=1E1) printf("00%u", usRD);
            else if (usRD>=1) printf("000%u", usRD);
                else printf("0000");
system("pause");
return 0;
}

```

执行结果截图：



```

请按照xxx-xxxx-xxxx的格式输入十一位手机号码： 135-6278-0001
输入的手机号码： 13562780001请按任意键继续. . . |

```

方案 3：使用字符串数据类型，逐位输入输出

```

#include<stdlib.h>
#include<stdio.h>
int main() {
    char cPhone[11];
    char cRound;
    char cBuffer;
    for (cRound=0; cRound<=10; ++cRound) {
        printf("请输入第%u位：", cRound+1);
        scanf("%c%c", &cPhone[cRound], &cBuffer);
    }
    printf("存储的手机号是：");
    for (cRound=0; cRound<=10; ++cRound) printf("%c", cPhone[cRound]);
    system("pause");
    return 0;
}

```

运行结果截图：

请输入第1位：8  
请输入第2位：2  
请输入第3位：3  
请输入第4位：8  
请输入第5位：3  
请输入第6位：8  
请输入第7位：2  
请输入第8位：8  
请输入第9位：0  
请输入第10位：1  
请输入第11位：0  
存储的手机号是：82383828010请按任意键继续...