Experiment1-董皓彧

环境:

```
Visual Stdio Code 1.83.0
gcc.exe (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0
```

必做题

必做题1

- 代码:

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

int main() {
    short a = 10;
    int b = 100;

    int short_length = sizeof a;
    int int_length = sizeof(b);
    int long_length = sizeof(long);
    int char_length = sizeof(char);

    printf("short=%d, int=%d, long=%d, char=%d\n",
        short_length, int_length, long_length, char_length);

    return 0;
}

```

- 输出:

```
short=2, int=4, long=4, char=1
```

- 运行截图:

- 结果分析:

`short`, `int`, `long`, `char` 数据类型分别占用 `2`, `4`, `4`, `1` 字节。`sizeof` 返回当前变量或数据类型所占内存大小。

必做题2

- 代码：

```
#include<stdio.h>

int main() {
    int m = 306587;
    long n = 28166459852;
    printf("m=%hd, n=%hd\n", m, n);
    printf("n=%d\n", n);

    return 0;
}

```

- 输出：

```
m=-21093, n=4556
n=-1898311220
```

- 运行截图：

- 结果分析：
  - `m`，`n` 分别是 `int`，`long` 型变量，每个最大占用 `32-bit`，最大可以表示 $2^{32}-1$，即 `2147483647`. 但初始化 `n` 时所赋的值大于该上限，故是将 `28166459852` 后 `32bit` 存入变量 `n` 中（实际上这是一个未定义行为）。而变量 `m` 是合法的。
  - 第一次输出时，由于限定以 `short` 类型输出，而 `m`，`n` 中存储的数据都超出了 `short` 范围，故输出结果其实是 `m`，`n` 后 `16bit` 的结果；第二次输出便正常输出 `n` 中存储的数据。

必做题3

- 代码：

```
#include<stdio.h>

int main() {

    printf("Please type in a digital:\n");
    char c = getchar(), endl = getchar();

    if(c>='0' && c<='9') {
        printf("The ascii of the given character is: %d\n", c);
        printf("The character before the given character is: %c\n", c-
1);
```

```
        printf("The character after the given character is: %c\n", c+1);
        return 0;
    }
    else {
        printf("The given character is not >=0 and <=9");
        return 1;
    }

}

```

- 输入1：

```
Please type in a digital:
5
```

- 输出1：

```
The ascii of the given character is: 53
The character before the given character is: 4
The character after the given character is: 6
```

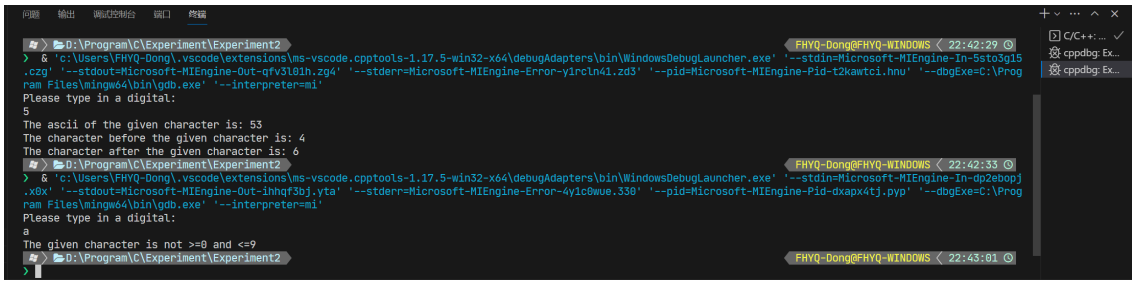
- 输入2：

```
Please type in a digital:
a
```

- 输出2：

```
The given character is not >=0 and <=9
```

- 运行截图：



选做题

选做题1

- 代码：

```
#include<stdio.h>

int main() {
    int x1, x2; // 32-bit 整数 2^31-1
```

```
    unsigned y; // 32-bit 无符号整数 2^32-1
    char c1, c2; // 8-bit 字符 根据以下部分代码增加变量c2

    x1 = 65535; // 2^16-1 < 2^31-1
    x2 = x1 + 5; // 2^16-1 + 5 < 2^31-1
    printf("enter y:\n"); // 揣摩原来代码的意图：输入y的值
    scanf("%d", &y); // 获取输入的值
    c1 = 97; c2 = 'A'; c2 = c2 + 32; // c1='a', c2=65, c2=65+32=97='a'

    printf("x1=%d, x2=%d, y=%d, c1=%c, c2=%c\n", x1, x2, y+15, c1, c2);
// 输出x1, x2, y+15, c1, c2的值

    return 0;
}

```

- 输入：

```
enter y:
24342
```

- 输出：

```
x1=65535, x2=65540, y=24357, c1=a, c2=a
```

- 运行截图：

选做题2

- 代码：

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define MAXN 100005 // 最长明文长度

char opt; // 操作 e:加密 d:解密
int K; // 密钥
int c_int[MAXN], m_int[MAXN]; // 密文整数数组，明文整数数组。// 担心溢出，所以用整数数组
char C[MAXN], M[MAXN]; // 密文，明文

int my_atoi(char c) { // 字符转整数
    return c - '0';

```

```
}

char my_itoa(int n) { // 整数转字符
    return n + '0';
}

void encrypt() { // 加密
    int len = strlen(M); // 明文长度
    for(int i=0; i<len; ++i) {
        m_int[i] = my_atoi(M[i]);
        c_int[i] = (m_int[i] + K) % ('~' - '0');
        C[i] = my_itoa(c_int[i]);
    }
    return;
}

void decrypt() { // 解密
    int len = strlen(C); // 密文长度
    for(int i=0; i<len; ++i) {
        c_int[i] = my_atoi(C[i]);
        m_int[i] = (c_int[i] - K) % ('~' - '0');
        while(m_int[i] < 0) m_int[i] += ('~' - '0'); // 防止负数
        M[i] = my_itoa(m_int[i]);
    }
    return;
}

int main() {
    printf("enter your option(e:encrypt d:decrypt):\n");
    scanf("%c", &opt); // 获取操作

    if(opt == 'e') {
        printf("enter your string to be encrypted(max:100000
characters):\n");
        scanf("%s", &M);
        printf("enter your key:\n");
        scanf("%d", &K); // 获取密钥

        encrypt(); // 加密

        printf("the encrypted string is:\n%s", C); // 输出密文
        return 0;
    }

    else if(opt == 'd'){
        printf("enter your string to be decrypted(max:100000
characters):\n");
        scanf("%s", &C); // 获取密文
        printf("enter your key:\n");
        scanf("%d", &K); // 获取密钥

        decrypt(); // 解密

        printf("the decrypted string is:\n%s", M); // 输出明文
        return 0;
```

```
    }

    else {
        printf("invalid option! Please enter 'e' or 'd'.\n");
        return 1;
    }

    return 0;
}

```

- 输入1:

```
enter your option(e:encrypt d:decrypt):
e
enter your string to be encrypted(max:100000 characters):
abcde
enter your key:
50
```

- 输出1:

```
the encrypted string is:
EFGHI
```

- 输入2:

```
enter your option(e:encrypt d:decrypt):
d
enter your string to be decrypted(max:100000 characters):
EFGHI
enter your key:
50
```

- 输出2:

```
the decrypted string is:
abcde
```

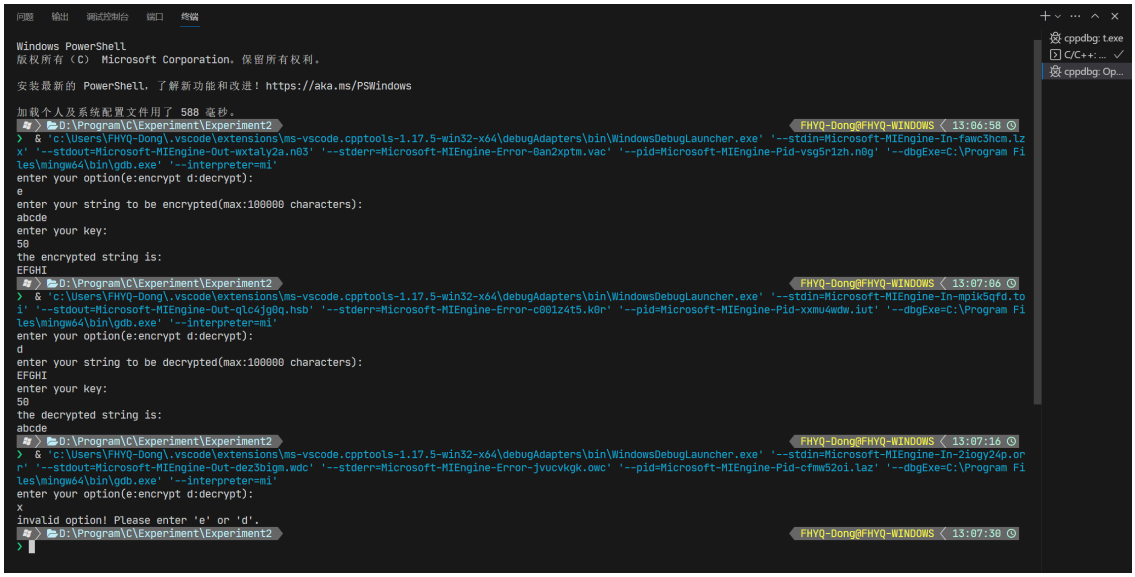
- 输入3:

```
enter your option(e:encrypt d:decrypt):
x
```

- 输出3:

```
invalid option! Please enter 'e' or 'd'.
```

- 运行截图:


问题 输出 调试控制台 端口 终端
cppdbg: t.exe
C/C++: ...
cppdbg: Op...
Windows PowerShell
版权所有（C） Microsoft Corporation。保留所有权利。
安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows
加载个人及系统配置文件用了 588 毫秒。
D:\Program\C\Experiment\Experiment2
FHYQ-Dong@FHYQ-WINDOWS 13:06:58
& 'c:\Users\FHYQ-Dong\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-fawc3hcm.lzx' '--stdout=Microsoft-MIEngine-Out-wxtaly2a.n03' '--stderr=Microsoft-MIEngine-Error-0an2xptm.vac' '--pid=Microsoft-MIEngine-Pid-vsg5r1zh.n0g' '--dbgExe=C:\Program Files\mingw64\bin\gdb.exe' '--interpreter=mi'
enter your option(e:encrypt d:decrypt):
e
enter your string to be encrypted(max:100000 characters):
abcde
enter your key:
50
the encrypted string is:
EFGHI
D:\Program\C\Experiment\Experiment2
FHYQ-Dong@FHYQ-WINDOWS 13:07:06
& 'c:\Users\FHYQ-Dong\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-mpik5qfd.toi' '--stdout=Microsoft-MIEngine-Out-qlc4jg0q.hsb' '--stderr=Microsoft-MIEngine-Error-c001z4t5.k0r' '--pid=Microsoft-MIEngine-Pid-xxmu4wdw.iut' '--dbgExe=C:\Program Files\mingw64\bin\gdb.exe' '--interpreter=mi'
enter your option(e:encrypt d:decrypt):
d
enter your string to be decrypted(max:100000 characters):
EFGHI
enter your key:
50
the decrypted string is:
abcde
D:\Program\C\Experiment\Experiment2
FHYQ-Dong@FHYQ-WINDOWS 13:07:16
& 'c:\Users\FHYQ-Dong\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-2iogy24p.or' '--stdout=Microsoft-MIEngine-Out-dez3bigm.wdc' '--stderr=Microsoft-MIEngine-Error-jvucvkgk.owc' '--pid=Microsoft-MIEngine-Pid-cfmw52oi.laz' '--dbgExe=C:\Program Files\mingw64\bin\gdb.exe' '--interpreter=mi'
enter your option(e:encrypt d:decrypt):
x
invalid option! Please enter 'e' or 'd'.
D:\Program\C\Experiment\Experiment2
FHYQ-Dong@FHYQ-WINDOWS 13:07:30