

### **Задача 1: Етапи на компилиране на C програми**

Създайте дадените по-долу **C** файлове. Компилирайте и изпълнете програмата с **GCC** като компилирате отделно двата файла стъпка по стъпка и след това ги свържете в изпълним файл **program**.

Използвайте текстови редактори за \*.i и \*.s файловете, както и hex редактор (напр. GHex) за да ги отворите и анализирате.

#### **File 1: program.c**

```
#include <stdio.h>

void swap(int*, int*);

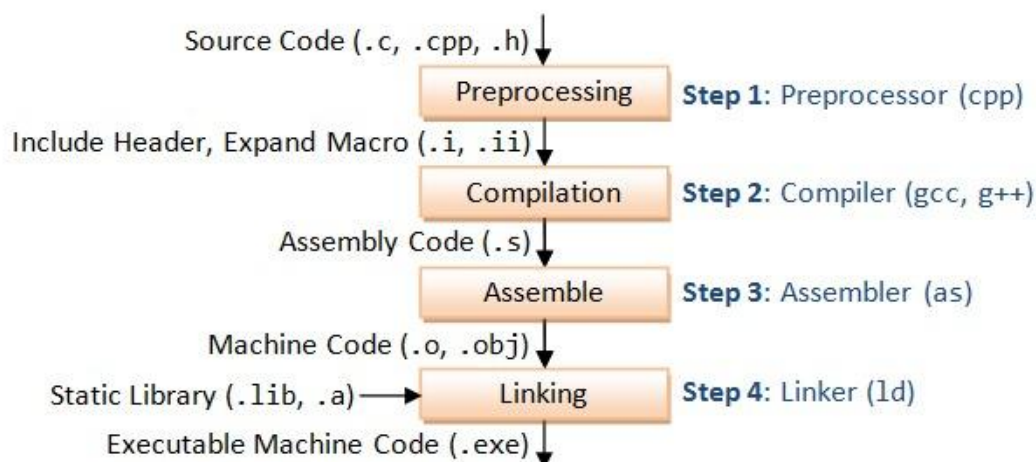
int main(void)
{
    int num1, num2;
    printf("Enter first number: ");
    scanf("%d", &num1);
    printf("Enter second number: ");
    scanf("%d", &num2);
    printf("Before swap: Num1 = %d, Num2 = %d\n", num1, num2);
    swap(&num1, &num2);
    printf("After swap: Num1 = %d, Num2 = %d\n", num1, num2);
    return 0;
}
```

#### **File 2: swap.c**

```
void swap(int* num1, int* num2)
{
    *num1 ^= *num2;
    *num2 ^= *num1;
    *num1 ^= *num2;
}
```

**Hint: Example for two file manual compilation and linking:**

Step	File 1	File 2
<b>Preprocessing</b>	gcc -E file1.c -o file1.i	gcc -E file2.c -o file2.i
<b>Compilation</b>	gcc -S file1.i -o file1.s	gcc -S file2.i -o file2.s
<b>Assemble</b>	gcc -c file1.s -o file1.o	gcc -c file2.s -o file2.o
<b>Linking</b>	gcc file1.o file2.o -o program	



## **Задача 2: Побитови операции (от работа в лекциите)**

Да се разработи програма тип меню, която обработва присъствия и отсъствия на клас от 32 студенти. **За целта да се използва 32-битова маска, а не масив или друга сложна структура от данни!**

- да се добави опция за въвеждане на присъствие;
- да се добави опция за въвеждане на отсъствие (изтриване на присъствие);
- да се добави опция за извеждане на номерата на студентите, които отсъстват;
- да се добави опция за извеждане на номерата на студентите, които присъстват;
- да се добави опция за промяна на статута на студент - ако присъства да се запише като отсъстващ или ако отсъства да се запише като присъстващ.

Да се добави валидация на данните - невалидна опция, невалиден номер на студент...

За решението на задачата може (но не е задължително) да се използва следната структура:

```

1. int main(void)
2. {
3.     uint32_t attendance = 0;
4.     int option;
5.     while (1)
6.     {
7.         printf("1. Set attendance\n");
8.         printf("2. Clear attendance\n");
9.         printf("3. Attendance info\n");
10.        printf("4. Change attendance\n");
11.        printf("5. Exit\n");
12.        scanf("%d", &option);
13.        if (option == 1)
14.        {
15.            // to do
16.        }
17.        else if (option == 2)
18.        {
19.            // to do
20.        }
21.        else if (option == 3)
22.        {
23.            // to do
24.        }
25.        // to do
26.    }
27.    return 0;
28.}

```

### **Задача 3: Конвертор на бройни системи**

Да се разработи конвертор на бройни системи. На първия ред да се въведе число, на втория основа на входната бройна система и на третия ред да се въведе основата на изходната бройна система.

Задължително е да направите валидация на входните данни.

Валидни основи за бройните системи да бъдат се зачитат от: 2 до 36. Валидни цифри са 0-9 и латинските букви a-z.

За ориентир вземете пример от този калкулатор:

<https://www.rapidtables.com/convert/number/base-converter.html>