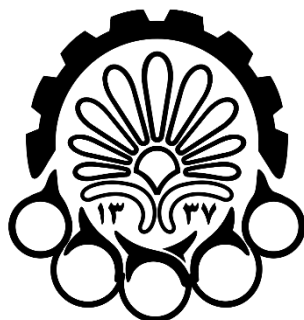


"به نام دادار داد گستر"



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

## کارگاه مبانی کامپیوتر و برنامه نویسی

عنوان

تمرین هشتم (C-Lab\_function)

مدرس

مهندس امیرحسین بابائیان

دانشجو

محمد یاراحمدی

۴۰۲۳۱۰۵۹

ترم پاییز ۰۳ - ۰۲

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

## فهرست

- پرسش اول (عدد فوق اول)..... ۱
- پاسخ پرسش اول ..... ۱
- پرسش دوم (فیوناچی)..... ۳
- پاسخ پرسش دوم ..... ۳
- پرسش سوم (تابع آکرمان)..... ۴
- پاسخ پرسش سوم ..... ۴
- پرسش چهارم (کافه)..... ۵
- پاسخ پرسش چهارم ..... ۵

## پرسش ۱

اگر ارقام عددی را از سمت راست جدا کنیم و به صورت یکرقمی، دو رقمی، سه رقمی و... بنویسیم، به طوری که تمام این ترکیبات جدا شده عدد اول باشند، آن عدد را عددی فوق اول می دانیم. به عنوان مثال ۱۷۳ عددی فوق اول است چون ۳ و ۷۳ و ۱۷۳ اول هستند. اگر یکی از این اعداد تفکیک شده عدد اول نباشد، در آن صورت بدیهیست که عدد دریافتی نیز فوق اول نیست! با توجه به این تعریف، برنامه ای بنویسید که عددی را به عنوان ورودی از کاربر دریافت کرده و فوق اول بودن یا نبودن عدد را مشخص کند.

## پاسخ پرسش ۱

برای حل این مسئله، ابتدا باید بدانیم که تعداد ارقام عدد ورودی چندتااست که اینکار را با ۲ تابع `size ۱` & `size ۲` انجام داده ایم. سپس اول بودن عدد ورودی را چک میکنیم. اگر اول بود آنگاه رقم سمت چپ عدد را حذف میکنیم و عملیات را با عدد جدید تا زمانی که به یک عدد اول تک رقمی برسیم، ادامه می دهیم. اگر تمام اعداد اول بودند آنگاه عدد ورودی اولیه سوپراول خواهد بود. در ضمن بدست آوردن تعداد ارقام به لین دلیل اهمیت دارد که بدانیم چند مرتبه عملیات حذف جایگاه سمت چپ باید انجام شود.

```
#include <stdio.h>
#include <math.h>
#include <stdbool.h>
int size (int n){
    int i = 10, cnt = 1;
    while(n % i != n){
        cnt ++;
        i *= 10;
    }
    return cnt;
}
int size2 (int n) {
    if (n < 10) return 1;
    return 1 + size2(n / 10);
}
bool aval (int n) {
    for (int i = 2; i <= sqrt(n); i++) if (n % i == 0) return false;
    return true;
}
bool superAval(int n) {
    if (n < 10) {
        return aval(n);
    }
}
```

```
        return aval(n) && superAval(n % (int)pow(10, size2(n) - 1));
    }
int main () {
    int n;
    scanf("%d", &n);
    printf("%d\n", superAval(n));
    return 0 ;}
```

## پرسش ۲

تابع فیبوناچی را به دو روش بازگشتی و مستقیم محاسبه کرده و این دو روش را در خروجی های بدست آمده با یکدیگر مقایسه کنید .

## پاسخ پرسش ۲

در روش بازگشتی اگرچه نوشتن کد آن کوتاه تر است ، اما چون هر بار جملات قبلی را دوباره محاسبه میکند به **cpu** فشار بیشتری وارد می شود و مدت زمان بیشتری طول می کشد که پاسخ نهایی را برایمان چاپ کند . راه حل رفع این مشکل این است که هر بار جملات بدست آمده را در یک آرایه ذخیره کنیم و جملات آرایه را صدا بزنیم .

در روش مستقیم برخلاف روش بازگشتی ، همه جملات دنباله را محاسبه نمی کند و فقط جمله مدنظر کاربر را بدست می آورد که اینکار علاوه بر صرفه جویی در زمان محاسبه ، به سیستم نیز فشار کمتری وارد می کند . پس گرچه روش بازگشتی راه حل مناسبی برای بعضی از مسائل می باشد اما برای تمام پرسش ها بهترین گزینه نیست !

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

int fibo(int n){
    if (n == 0 || n == 1) return 1 ;
    return fibo(n - 1) + fibo (n - 2) ;
}

int fibo2(int n){
    return (pow(((1 + sqrt(5))/2) , n ) - pow(((1 - sqrt(5))/2) , n )) / sqrt(5)
;
}

int main (){
    int n ;
    scanf("%d" , &n);
    printf("%d" , fibo2(n));
    return 0 ;
}
```

## پرسش ۳

برنامه ای بنویسید که با دریافت عدد  $n$  و  $m$  از کاربر ، تابع آکرمان این ۲ عدد را بصورت بازگشتی محاسبه کند .

## پاسخ پرسش ۳

طبق راهنمایی های دستور کار و ضابطه بازگشتی خود تابع داریم :

```
#include <stdio.h>
int acerman(int m , int n){
    if (m == 0 ) return n + 1 ;
    else if (m > 0 && n == 0) return acerman( m - 1 , 1 );
    else if (m > 0 && n > 0 ) return acerman ( m - 1 , acerman(m , n - 1));
}
int main (void){
    int m , n ;
    scanf("%d%d" , &m , & n ) ;
    // acerman(m , n);
    printf("%d" , acerman(m,n));
    return 0 ;
}
```

## پرسش ۴

برنامه ای بنویسید که کافه داده شده در دستور کار را مدیریت کند و از کاربر سفارش بگیرد و هر بار موجودی کافه را نمایش دهد.

## پاسخ پرسش ۴

در ابتدا، موجودی ها را با دستور **define** بصورت گلوبال تعریف میکنیم و همینکار را در تابع خرید بصورت **static** انجام می دهیم زیرا قرار است بعد از هر بار خرید، مقدار موجودی ما تغییر کند. برای سفارش گیری هم از دستور **switch-case** استفاده کرده ایم که مطابق با منوی کافه ورودی را از کاربر بگیرد. کامل شده کد دستور کار مطابق زیر می باشد:

```
#include <stdio.h>
#include <stdbool.h>
#define init_cake_num 40
// Write your code here
#define init_coffee_num 40
#define init_cookie_num 40
/* global variable declaration */
// Write your code here
// int init_cookie_num , init_coffee_num ;
int choice ;
void menu(int cake, int cookie, int coffee) {
    printf("What do you want to order?\n");
    printf("1) cake : %d\n", cake);
    printf("2) cookie : %d\n", cookie);
    printf("3) coffee : %d\n", coffee);
}

void buy() {

    scanf("%d", &choice);

    /* local variable declaration */
    int pic = 0 ;

    /* static variable declaration */
    // Write your code here
    static int cake = init_cake_num ;
    static int coffee = init_coffee_num ;
    static int cookie = init_cookie_num ;

    switch (choice) {
        case 1:
            // Write your code here
```

```

        cake -- ;
        menu(cake, cookie , coffee);
        break ;
    case 2:
        // Write your code here
        cookie -- ;
        menu(cake, cookie , coffee);
        break ;
    case 3:
        // Write your code here
        coffee -- ;
        menu(cake, cookie , coffee);
        break ;
    case 5:
        printf("Do you want a picture?\n1) yes\n0) no\n");
        int a;
        scanf("%d", &a);
        if (a)
            pic++;
        break;
    default :
        menu(cake, cookie , coffee);
        break ;
}

printf("pic num = %d\n", pic);
}

int main() {

    menu(init_cake_num, init_cookie_num, init_coffee_num);

    while(choice != 5)
        buy();

    printf("C U soon\n");

    return 0;
}

```