

آزمون پایانی مبانی کامپیوتر و برنامه‌سازی

پرهام الوانی

نیم سال دوم ۹۹-۱۳۹۸

۱ پیش درآمد

۱.۱ تعهدنامه

نام درس: مبانی کامپیوتر و برنامه‌سازی
نام استاد: پرهام الوانی
شماره دانشجویی:
تاریخ آزمون: ۲۱ تیر ۱۳۹۹

من،، به عنوان یک عضو فرهیخته از جامعه انسانی،
تعهد مینمایم که اصول و قواعد متعارف در پاسخدهی شرافتمندانه به سوالات این امتحان
را رعایت نمایم. به طور مشخص:

- از هیچ فردی (خواه در این کلاس و خواه در خارج از کلاس) جهت پاسخدهی به تمام
یا بخشی از این امتحان کمک نمیگیرم و نیز به فرد دیگری کمک نخواهم کرد.
- زمان در نظر گرفته شده برای امتحان را دقیقاً رعایت خواهم کرد.

امضا

۲.۱ نکات

- سوالات آزمون را می‌توانید به صورت تایپ شده یا دستی تحویل دهید.
- برای جلوگیری از هرگونه خطا یک نسخه از پاسخهای خودتان را در قالب آرشو برای
من ایمیل کنید.
- در صورت هرگونه پرسش یا ابهام از سامانه Adobe Connect استفاده کنید.
- پاسخ به ۲۰ نمره از آزمون کفایت می‌کند و اضافهی آن نمرهی امتیازی خواهد بود.

۲ عدد تصادفی n رقمی

برنامه‌ای بنویسید که با دریافت عدد n یک عدد تصادفی n رقمی تولید کند.
برای درک بهتر از عملکرد تابع به مثال‌های زیر توجه کنید:

Input:

2

Output:

21

Input:

5

Output:

12345

۶ نمره

نکات در نظر داشته باشید خروجی شما لزوماً در قالب `int` جا نخواهد شد بنابراین می‌بایست از `char[]` برای نگهداری آن استفاده کنید.

۳ فیبوناچی در فایل

تابع زیر به شکلی تکمیل کنید که جمله ی n ام فیبوناچی را محاسبه کرده و آن را در انتهای فایل داده شده یادداشت کند. در نظر داشته باشید عدد به صورت متنی نوشته می شود.

```
void fib_in_file(char *name, int n) {  
    // your code goes here  
}
```

برای درک بهتر از عملکرد تابع به مثال های زیر توجه کنید:

Input: 1
Output: 1

Input: 2
Output: 1

Input: 3
Output: 2

۴ نمره

نکات در نظر داشته باشید که حتما در هنگام باز کردن فایل خطا را نیز مورد بررسی قرار دهید.

۴ دانشجویان مرتب

تابع زیر را به شکلی تکمیل کنید که با دریافت آرایه‌ای از دانشجویان آن را به صورت مرتب شده بازگرداند. مرتب‌سازی می‌بایست براساس شماره دانشجویی و با استفاده از تابع *qsort* صورت بپذیرد. اگر به نظرتان امضا تابع نیاز به تغییر دارد با ذکر علت آن را تغییر دهید.

```
struct student {
    int grade;
    int id;
    char name[200];
};

void students_sort(struct student *students,
    int students_len) {
    // your code goes here
}
```

۴ نمره

۵ رشته بزرگ کردن

تابع زیر یک رشته متشکل از حروف بزرگ، کوچک و اعداد انگلیسی را گرفته و تمامی حروف آن را بزرگ می‌کند. نتیجه نهایی در قالب یک رشته برای کاربر بازگشت داده می‌شود. در نظر داشته باشید که رشته‌ی ورودی کاربر نباید عوض شود. رشته‌ی خروجی می‌بایست حافظه‌ای دقیقاً برابر با اندازه رشته‌ی اولیه اشغال کند.

```
char *to_upper(char *input) {  
    // your code goes here  
}
```

برای درک بهتر از عملکرد تابع به مثال‌های زیر توجه کنید:

Input: aAaA
Output: AAAA

Input: abcd
Output: ABCD

Input: aaBB
Output: AaBB

Input: aaBB10a
Output: AaBB10A

۴ نمره

۶ درباره لیست پیوندی

۱. خروجی برنامه را ذکر کنید و دلیل تغییر قسمت before و after را بنویسید.
۲. شکل کلی لینک لیست را رسم کنید و تغییر اعمال شده را با رنگ دیگری نمایش دهید.
۳. چه اتفاقی برای نود سوم می افتد و برنامه نویس باید به چه نکاتی دقت کند؟ (صرفاً توضیح دهید)

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

struct Node{
    char data;
    struct Node *next;
};

int main()
{
    struct Node* n1 = malloc(sizeof(struct Node));
    struct Node* n2 = malloc(sizeof(struct Node));
    struct Node* n3 = malloc(sizeof(struct Node));

    n1->data = 'a';
    n1->next = n2;

    n2->data = 'b';
    n2->next = n3;

    n3->data = 'c';
    n3->next = NULL;

    printf("before\n");

    struct Node *temp = n1;
    while( temp != NULL){
        printf("data:_%c\n", temp->data);
        temp = temp->next;
    }

    // the change
    n1->next->next = NULL;
```

```
printf("after\n");  
temp = n1;  
while( temp != NULL){  
    printf("data:_%c\n", temp->data);  
    temp = temp->next;  
}  
  
return 0;  
}
```

٦ نمبرہ