



به نام خدا  
دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران  
مبانی کامپیوتر و برنامه‌نویسی



استاد : دکتر مرادی، دکتر هاشمی

تمرین سری ۶  
موعد تحویل: سه شنبه ۶ دی

نیمسال اول ۱۴۰۱-۰۲

**\*تمرین‌های تئوری**

۱. الف) توضیح دهید خروجی قطعه کد زیر چه خواهد بود. فرض کنید آدرس خانه  $a[0]$  برابر  $x1000$  است.

```
int main(){
    int* a;
    a = (int*)malloc(100*sizeof(int));
    for (int i = 0; i < 100; i++)
        a[i] = 3 * i + 1;
    printf("%d\n", a);
    printf("%d\n", (a+4));
    printf("%d\n", *(a+4));
    printf("%d\n", a[99]);
    printf("%d\n", &a[99]);
    printf("%d\n", &a[99]+2);
}
```

ب) تفاوت تابع `malloc` و `calloc` را توضیح دهید.

۲. با توجه به مفاهیم تخصیص حافظه‌ی پویا در زبان C مشکل کد زیر را توضیح دهید و بگویید چطور آن را می‌توان حل کرد.

```
#define true 1
int main(){
    int* a;
    while (true){
        a = (int*)malloc(100*sizeof(int));
        for (int i = 0; i < 100; i++){
            scanf("%d",&num);
            *(a+i) = num;
            for (int j = 0; j < i; j++){
                if (num == a[i])
                    a[i]++;
            }
            printf("%d\n", a[i]);
        }
    }
}
```

۳. توضیح مختصری دهید که هر یک از توابع زیر چه کاری انجام می‌دهند. (برای سادگی می‌توانید با یک لیست پیوندی دلخواه مانند

{1 → 2 → 5 → 8 → 4 → 2 → 5 → 3} کار کنید ولی در نهایت بگویید به صورت کلی تابع چه کاری انجام می‌دهد.)

```
typedef struct Node{
    int data;
    struct Node* next;
} Node;
```

(الف)

```
Node* g(Node* head){
    if (head == NULL || head->next == NULL)
        return head;
    Node* prev = NULL;
    Node* cur = head;
    while (cur->next){
        prev = cur;
        cur = cur->next;
    }
    prev->next = NULL;
    cur->next = head;
    head = cur;
    return head;
}
```

(ب)

```
Node* f(Node* head){
    if (!head)
        return head;
    Node* next = head->next;
    head->next = NULL;
    while (next){
        Node* new_next = next->next;
        next->next = head;
        head = next;
        next = new_next;
    }
    return head;
}
```

ج) (امتیازی)

```
void p(Node* head){
    if (head == NULL)
        return;
    p(head->next);
    printf("%d\t", head->data);
    return;
}
```

### \*تمرین کامپیوتری

۴. فرض کنید ساختار زیر را در اختیار داریم:

```
typedef struct Point{
    int x;
    int y;
} Point;
```

الف) ساختاری به اسم Node درست کنید که یک خانه لیست پیوندی از جنس Point باشد.

ب) تابعی بنویسید که آرایه‌ای از جنس Point\* و طولش را دریافت کند و سپس آن را در یک لیست پیوندی ذخیره کند و پوینتر به عضو اول لیست پیوندی را خروجی دهد.

راهنمایی: برای تعریف هر node جدید در لیست پیوندی از حافظه‌ی پویا استفاده کنید. (مانند کد پایین)

```
Node* node = (Node*)malloc(sizeof(Node));
node-> ... = ...;
```

تعریف تابع به این شکل باشد:

```
Node* convert(Point* arr[1000], int size);
```

ج) (امتیازی) تابعی بنویسید که فاصله‌ی نقاط مجاور در یک لیست پیوندی از جنس Point را چاپ کند.

مثال: اگر لینکد لیست به صورت: (0,0)->(3,4)->(8,16) باشد، مقادیر 5 و 13 چاپ می‌شود.

تعریف تابع به این شکل باشد:

```
void print_distance(Node* head);
```

## ۵. (امتیازی)

الف) ساختاری (struct) به اسم Student درست کنید که شامل اسم دانشجو، شماره‌ی دانشجویی و معدلش باشد.

ب) برنامه‌ای بنویسید که ابتدا یک عدد به عنوان تعداد احتمالی دانشجویها دریافت کند و سپس آرایه‌ای پویا از جنس ساختار بخش قبل با طول داده شده درست کند. سپس اطلاعات دانشجویها را دریافت کند تا وقتی که شماره‌ی دانشجویی داده شده مقدارش 0 باشد. در پایان اسم دانشجویها را به ترتیب صعودی معدلشان چاپ کند. (دقت کنید که ممکن است مقدار دانشجویهایی که اطلاعات آنها را می‌دهیم از عدد اولیه‌ی داده شده بیشتر شود. در این صورت سائز آرایه‌ی پویا را با تخصیص مجدد حافظه دو برابر کنید).

مثال:

ورودی:

2

Ali 1 18

Hossein 2 20

Reyhane 3 19

Hoda 4 16

End 0 0

خروجی:

Hoda

Ali

Reyhane

Hossein

بعد از ورودی سوم هم باید آرایه‌ی پویا سائزش دو برابر شود.

## \*توضیحات

۱. فرمت فایل نهایی که آپلود می‌کنید باید به صورت HW6-SID.zip باشد که داخل آن فایل‌های C و یک فایل pdf برای تمرین‌های تئوری باشد. (برای هر سوال تمرین کامپیوتری یک فایل C مجزا داشته باشید) در نام فایل به جای عبارت SID باید شماره دانشجویی شما باشد. برای مثال اگر کسی شماره دانشجویی اش ۸۱۰۱۰۱۲۲۲ باشد، نام فایل باید HW6-810101222.zip باشد.

۲. توجه داشته باشید که به جز سوال ۵ در فایل‌هایی که برای سوال کامپیوتری ارسال می‌کنید، نباید تابع main داشته باشند و فقط باید توابع را با نام‌های گفته شده در سوال بسازید.

۴. تمیز بودن کد شما اهمیت ویژه‌ای دارد. عدم رعایت فاصله از سر خط در کدنویسی و نامگذاری نامناسب متغیرها و توابع تا ۱۵٪ نمره‌ی آن سوال نمره‌ی منفی دارد.

۵. به فرمت خروجی (عبارت چاپ شده) خواسته شده در هر سوال توجه کنید. در صورتی که از فرمت گفته شده استفاده نکرده باشید نمره‌ی آن را نخواهید گرفت. دقت کنید که به غیر از مواردی که در صورت سوال یا شکل به آن‌ها اشاره شده مورد دیگری را چاپ نکنید.

۶. مجاز به استفاده از دستور goto و متغیرهای global نیستید. در صورت استفاده در هر سوال، نمره‌ی آن سوال صفر در نظر گرفته می‌شود.