

سوال ۱

خروجی قطعه کد زیر چیست؟ نحوه تولید خروجی را با ترسیم شکل حافظه شرح دهید. (۲۰ نمره)

```
int j = 0, i = 0, *p1, **p2, **p3;
p2 = p3 = (int **)calloc(3, sizeof(int *));
for(i = 0; i < 3; i++){
    j = 2*i + 1;
    *p2 = (int *)malloc(j * sizeof(int));
    **p2 = j;
    p1 = *p2;
    while(j > 1){
        p1++;
        *p1 = j + *(p1 - 1);
        j--;
    }
    p2++;
}
for(i = 2; i >= 0; i--)
    for(j = 0; j < 2 * i + 1; j++)
        printf("[%d][%d] = %d\n", i, j, *((*(p3 + i))+j));
```

سوال ۲

تابع زیر را تکمیل کنید. این تابع اشتراک دو مجموعه a و b را محاسبه می‌کند. تعداد اعضای نتیجه با `size_result` مشخص می‌شود و اشاره‌گر به مجموعه نتیجه `return` می‌شود. دقت کنید که نمی‌توانید آدرس شروع یک آرایه را `return` کنید چون آرایه یک متغیر `automatic` است که در `stack` قرار می‌گیرد و با تمام شدن تابع از بین می‌رود. یک برنامه بنویسید که اطلاعات دو مجموعه (ابتدا تعداد اعضا و سپس خود اعضا) را از کاربر بگیرد و با این تابع اجتماع آنها را محاسبه کرده و نتیجه را چاپ کند. (۲۰ نمره)

```
float * intersection(float *a, int size_a, float *b, int size_b, int *size_result)
```

ورودی

```
<n1>
<x_1> <x_2> <...> <x_n1>
<n2>
<y_1> <y_2> <...> <y_n2>
```

خروجی

```
<n3>
<z_1> ... <z_n3>
```

سوال ۳

تابع زیر را تکمیل کنید. این تابع یک آرایه از رشته (تعداد رشته‌ها برابر rows و حداکثر طول هر رشته برابر len است و همه اعضای آن حروف کوچک است) و یک آرایه از struct count را به عنوان ورودی می‌گیرد، سپس این رشته‌ها را بررسی کرده و تعداد دفعات تکرار هر کلمه را پیدا کرده و آنها را در آرایه result ذخیره کرده و در نهایت تعداد کلمات غیر تکراری را بر میگرداند.

```
struct count {
    char word[20];
    int cnt;
};
int frequency(int rows, int len, char strings[rows][len], struct count result[]);
```

برای مثال اگر رشته‌ها به شکل زیر باشد

```
this is a test string
test string is for test
test is test
```

این تابع عدد ۶ را برمیگرداند و آرایه result به شکل زیر خواهد بود

```
{this 1} {is 3} {a 1} {test 5} {string 2} {for 1}
```

برنامه‌ای بنویسید که در ابتدا rows و len را از کاربر بگیرد، بعد از آن rows رشته را از کاربر بگیرد و با استفاده از تابع فوق تعداد کلمات غیر تکراری، خود کلمات و دفعات تکرار آنها را بنویسید. (۲۰ نمره)

ورودی

```
<rows> <len>
<abc .....>
<abc .....>
```

خروجی

```
<unique #>
<abc> <xyz>
...
<abc> <xyz>
```

سوال ۴

فرض کنید یک لیست پیوندی با استفاده از `struct` زیر از نقاط فضای دو بعدی ایجاد شده است. قصد داریم در این لیست نقاطی که مقدار `x` یکسانی دارند را حذف کنیم. اگر چندین نقطه وجود دارد که مقدار `x` یکسانی دارند، تنها نقطه‌ای در لیست باقی میماند که مقدار `y` از بقیه نقاط با `x` یکسان کمتر باشد. تابعی برای این منظور بنویسید. توجه کنید که اجازه تولید یک لیست جدید را نداریم.

```
struct point {
    int x;
    int y;
}
struct node {
    struct point data;
    struct node *next;
}
```

برنامه‌ای بنویسید که مختصات ۱۰ نقطه را از کاربر بگیرد، سپس به روش فوق نقاط تکراری را حذف کرده و نتیجه را چاپ کند. (۲۰ نمره)

ورودی

```
<x1> <y1>
...
<x10> <y10>
```

خروجی

```
<x1> <y1>
...
<xn> <yn>
```

سوال ۵

برای نگهداری زمان از `struct` زیر استفاده می‌شود.

```
struct time {
    int hour;
    int min;
    int sec;
    int mili_sec;
};
```

در این `struct` `mili_sec` بین ۰ و ۹۹۹، `sec` بین ۰ و ۵۹، `min` بین ۰ و ۵۹ و `hour` یک مقدار غیر منفی است. فرض کنید اطلاعات زمان در یک فایل `Binary` ذخیره شده است. برنامه‌ای بنویسید که اسم فایل را از کاربر بگیرد و میانگین زمان‌ها، بزرگترین و کوچکترین زمان را چاپ کند. (۲۰ نمره)

ورودی

<filename>

خروجی

<average>

<max>

<min>