نام درس: کارگاه مبانی کامپیوتر و برنامه‌نویسی (زبان C++)	پروژه برنامه نویسی

تمرین شماره۱:

مثلث خیام یک آرایه سه گوش از اعداد به شکل زیر است:

```
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5  10  10   5   1
    1   6  15  20  15   6   1
```

هر عدد در مثلث خیام، یکی از ترکیبات c(n,k) است. اگر سطر ها و ستون ها را از صفر شروع کنیم، عدد واقع در ردیف n و ستون k برابر c(n,k) است. به عنوان مثال، c(6,2)=15 که در سطر ۶ ام و ستون ۲ ام قرار دارد. برنامه ای بنویسید که مثلث خیام دوازده سطری چاپ کند.

تمرین شماره ۲:

تابع ()average را که **میانگین پنج عدد** را برمی گرداند را بنویسید.

تمرین شماره ۳:

تابع ComputeCircle() که مساحت و محیط یک دایره را محاسبه و نشان می دهد را پیاده سازی کنید. این تابع باید شعاع دایره را از ورودی دریافت کرده و محیط و مساحت آن را محاسبه کند و در صفحه نمایش نشان دهد.

تمرین شماره ۴:

برنامه ای برای جست و جوی خطی بنویسید که به جای برگرداندن اولین محل قرار گرفتن عنصر کلید در آرایه، آخرین محل قرار گرفتن آن در آرایه را برگرداند.

۱۹	۵۶	۱۲	۴۸	۵۶	۲۳

به عنوان مثال، در آرایه بالا، اگر عنصر کلید (عنصری که به دنبال آن در آرایه می گردیم) عدد ۵۶ باشد، برنامه ای که می نویسید باید به جای برگرداندن ایندکس ۱ (مربوط به اولین محل عنصر کلید)، ایندکس ۴ را برگرداند.

تمرین شماره ۵:

تابع زیر را که یک مقدار را از آرایه حذف می کند را بنویسید:

void remove (float a[], int& n, int i);

تابع بالا باید به گونه ای نوشته شود که اگر به عنوان مثال، عنصر a[i] را حذف می کند که تمام عناصر بعد از آن را یک خانه به عقب حرکت می دهد (ایندکس خانه های بعد از آن خانه را یک واحد کاهش می دهد) و n را یک واحد کاهش می دهد.

تمرین شماره ۶:

تابعی بنویسید که از اشاره گرها برای جست و جوی آدرس (ایندکس) یک عدد صحیح مفروض(کلید) در یک آرایه استفاده کند. اگر عدد مفروض (کلید) پیدا شود، تابع، آدرس (ایندکس) آن را برگرداند و در غیراینصورت null را برگرداند.

تمرین شماره ۷:

کد تابع زیر را بنویسید. این تابع حداکثر n بایت با شروع از s2* را با بایت های متناظر با شروع از s1* مقایسه می کند که n تعداد بایت هایی است که s2 می تواند افزایش یابد قبل از اینکه به کاراکتر '\0' اشاره کند. اگر همه n بایت معادل باشند، تابع باید 0 را برگرداند. درغیراینصورت بسته به اینکه در اولین اختلاف بایت موجود در s1 کوچک تر یا بزرگ تر از بایت موجود در s2 باشد، مقدار 1- یا 1 را برگرداند.

int cmp(char* s1,char* s2);

موفق باشید