

۱- تابعی به نام `sum_pos` بنویسید که وظیفه آن برگرداندن حاصلجمع همه عناصر یک ماتریس پنج در شش باشد که به آن داده می شود.

۲- تابعی به نام `inv` بنویسید که رشته ای را که بعنوان پارامتر ورودی می گیرد، بطور معکوس چاپ کرده و تعداد فضاهای خالی آن را برگرداند.

۳- تابع یا توابعی به نام `add` بنویسید که بتواند یک/دو/ یا سه ورودی از جنس عدد صحیح داشته باشد و حاصلجمعشان را چاپ کند یا ورودی رشته ای داشته باشد و آنها را بدنبال هم زنجیرکرده و نمایش دهد (اگر رشته دوم داده نشود بجای آن نقطه بگذارد)

۴- تابعی بنام `swap` بنویسید که مقادیر دو متغیر اعشاری را که بعنوان پارامتر ورودی گرفته است با هم عوض کند.

۵- تابعی به نام `sort` برای مرتب سازی صعودی آرایه های عددی اعشاری بنویسید. این تابع آدرس شروع آرایه را به همراه تعداد عناصرش می گیرد. ضمناً می توانید از تابع `swap` هم کمک بگیرید.

۶- برنامه ای بنویسید که ابتدا تعداد عناصر یک آرایه اعشاری پویا و سپس اعداد مربوطه را گرفته و آنها (به کوک تابع `sort`) مرتب کرده و نمایش دهد.

۷- کلاسی بنام `time` برای توصیف زمان بنویسید با خصوصیات زیر:

❖ متغیرهای عضوی عمومی برای بیان ساعت، دقیقه، ثانیه به نامهای `h`، `m` و `s` داشته باشد.

❖ دارای سازنده ای باشد که در صورت عدم مقدار دهی هر سه عضو مربوطه را صفر دهد و ضمناً قادر باشد به هر سه شکل: فقط ساعت/ فقط ساعت و دقیقه/ ساعت - دقیقه - ثانیه مقادیر را دریافت نماید.

❖ دارای توابع عضوی به نامهای `show` و `get` برای نمایش زمان روی مونتور یا دریافت آن از صفحه کلید باشد.

❖ تابع عضوی به نام `seconds` داشته باشد که نتیجه تبدیل زمان به ثانیه را برگرداند.

❖ تابع عضوی به نام `compare` برای مقایسه کردن دو زمان داشته باشد. (اگر زمان اصلی از زمان دوم کوتاهتر بود منفی یک، اگر بلندتر بود مثبت یک و اگر مساوی بودند صفر برگرداند)

❖ تابع عضوی به نام `add` برای جمع کردن زمان اصلی با زمان دیگر که به آن داده می شود داشته باشد که اصلاحات لازم را نیز پس از جمع لحاظ کند.

۸- برنامه ای بنویسید که صد زمان (`time`) را گرفته و در نهایت بلندترین زمانها را به همراه حاصلجمع همه زمانهای وارد شده، نمایش دهد.

← جوابها از صفحه بعد

```

#include <string.h>
#include <iostream.h>

double sum_pos(double x[5][6])           //----- 1 -----
{
    double s = 0;
    int i, j;
    for(i = 0; i < 5; i++)
        for(j = 0; j < 6; j++)
            if(x[i][j] > 0)
                s += x[i][j];
    return s;
}

int inv(char p[])                         //----- 2 -----
{
    int i, n = 0;
    for(i = strlen(p) - 1; i >= 0; i--)
    {
        cout << p[i];
        if(p[i] == ' ')
            n ++;
    }
    return n;
}

void add(int a, int b = 0, int c = 0)     //----- 3 -----
{
    int d = a + b + c;
    cout << d;
}

void add(char p[], char q[] = ".")       //----- 3 -----
{
    strcat(p, q);
    cout << p;
}

void swap(double& a, double& b)          //----- 4 -----
{
    double t;
    t = a;
    a = b;
    b = t;
}

```

```
void sort(double *x, int n)           //----- 5 -----
{
    int i, j;
    for(i = n - 1; i > 0; i--)
        for(j = 0; j < i; j++)
            if(x[j] > x[j + 1])
                swap(x[j], x[j + 1]);
}
```

```
void main()                           //----- 6 -----
{
    double *u;
    int m, i;
    cin >> m;
    u = new double[m];
    for(i=0; i<m; i++)
        cin >> u[i];

    sort(u, m);
    for(i=0; i < m; i++)
        cout << u[i] << "\n";
}
```

```
//----- 7 -----
```

```
#include <iostream.h>
```

```
class time
```

```
{
```

```
public:
```

```
    int h, m, s;
```

```
    time(int hh = 0, int mm = 0, int ss = 0)
```

```
    {
```

```
        h = hh;
```

```
        m = mm;
```

```
        s = ss;
```

```
    }
```

```
    void show()
```

```
    {
```

```
        cout << h << ":" << m << ":" << s << "\n";
```

```
}

void get()
{
    cin >> h >> m >> s;
}

int seconds()
{
    return h * 3600 + m * 60 + s;
}

int compare(time t)
{
    if(seconds() > t.seconds())
        return 1;

    if(seconds() < t.seconds())
        return -1;

    return 0;
}

void add(time t)
{
    s += t.s;
    m += t.m;
    h += t.h;
    if(s >= 60)
    {
        s -= 60;
        m ++;
    }
    if(m >= 60)
    {
        m -= 60;
        h ++;
    }
}

};

//----- 8 -----

void main()
{
```

```
time x[100], max, sum;
int i;
for(i=0; i<100; i++)
{
    x[i].get();
    sum.add(x[i]);
    if(x[i].compare(max) == 1)
        max = x[i];
}
cout << "sum = ";
sum.show();
cout << "max = ";
max.show();
}
```