الهام اسدی
۹۷۱۲۲۵۹۲۰۷

آزمون دوم برنامه نویسی دو

۱. یک تابع برای بدست اوردن تعداد روز هایی که تا یک ماه مشخص می گذرد داریم به نام *days*. با کمک آن در تابع *distance* فاصله بین دو تاریخ را بدست می آوریم. تنها مورد قابل توجه این است که شش ماه اول سال ۳۱ روزه ولی بقیه ۳۰ روزه هستند.

```cpp
#include <iostream>
using namespace std;

struct Date {
    int year;
    int month;
    int day;
};

typedef struct Date Date;

int days(int month) {
    int d = 0;

    if (month > 6) {
        d += 6 * 30;
        month -= 6;
    } else {
        d += month * 30;
        month = 0;
    }

    d += month * 31;

    return d;
}

int distance(Date from, Date to) {
    return (
        (to.year - from.year) * 365 +
        (days(to.month) - days(from.month)) +
        (to.day - from.day)
    );
}

void getDate(Date &date) { cin >> date.year >> date.month >> date.day; }

int main() {
    Date now, birth;
    cout << "Enter today's date: (Format: Y M D)" << endl;
    getDate(now);
    cout << "Enter the birth date: (Format: Y M D)" << endl;
    getDate(birth);
    cout << "The days have been spent from the birth date: " << distance(birth
, now) << endl;
}
```

الهام اسدی
۹۷۱۲۲۵۹۲۰۷

۲. کد به صورت زیر است:

```cpp
#include <iostream>
using namespace std;

enum Color {
    red = 0,
    blue = 1,
    yellow = 2
};

typedef struct Team {
    int code;
    string name;
    int foundation_year;
    int color;
} Team;

void getTeam(Team &team) {
    cin >> team.code >> team.name >> team.foundation_year >> team.color;
}

string colorName(int color) {
    switch (color)
    {
    case red:
        return "red";
    case blue:
        return "blue";
    case yellow:
        return "yellow";
    default:
        return "no color";
    }
}

void showTeam(const Team &team) {
    cout << "Team code: " << team.code << endl;
    cout << "Team foundation year: " << team.foundation_year << endl;
    cout << "Team color: " << colorName(team.color) << endl;
}

int main() {
    int n;

    cout << "Enter the number of teams: " << endl;
    cin >> n;

    Team *teams = new Team[n];
```

```
    for(int i = 0; i < n; i++) {
        cout << "Enter the " << i << "-
th team info: (Format: code name foundation_year color)" << endl;
        cout << "Colors = [ 0: red, 1: blue, 2: yellow ]" << endl;
        getTeam(teams[i]);
    }

    for(int i = 0; i < n; i++) {
        cout << "The " << i << "-th team info: " << endl;
        showTeam(teams[i]);
    }
}
```

که در آن در ابتدا n را از ورودی گرفته و سپس به اندازه آن یک آرایه ساخته می شود تا اطلاعات تیم ها در آن ذخیره شود.
سپس در یک حلقه ورودی گرفته می شود و سپس در حلقه ای دیگر اطلاعات تیم ها به نمایش در میایند. برای نگه داری رنگ پیراهن هر تیم از $enum$ کمکی استفاده کردیم تا در خروجی بتوانیم رشته مربوط به آن را نمایش دهیم.

۳. در نظر گرفتن حالاتی که باعث ایجاد خطا در برنامه شده را می توان با پردازش استثنا کنترل کرد و به ما امکان می دهد به آن رسیدگی کنیم برای مثال به کاربر نمایش خطا دهیم و یا برای رفع مشکل اقدام کنیم و یا حتی برای تست کردن برنامه می تواند مفید باشد.

```cpp
#include <iostream>

using namespace std;

int main() {
    cout << "Enter two number to calculate a / b." << endl;
    cout << "(if b is zero then an exception will raise)" << endl;

    int a, b;
    cin >> a >> b;

    try {
        if (b == 0) {
            throw "Divison by zero!!";
        }
        cout << (float)a/b << endl;
    } catch(const char* err) {
        cout << err << endl;
    }
}
```

این کد تقسیم بر صفر را نمایش می دهد که اگر مخرج صفر باشد یک استثنا می باشد.

الهام اسدی
۹۷۱۲۲۵۹۲۰۷

۴. کد زیر

```cpp
#include <iostream>
#include <cstring>

using namespace std;

int main() {
    char *s = new char[256];
    char *t = new char[256];

    cin >> s >> t;

    char *r = new char[512];

    for(int i = 0; i < strlen(s) + strlen(t); i++) {
        r[i] = (i < strlen(s) ? s[i] : t[i - strlen(s)]);
    }

    cout << r << endl;
}
```

در ابتدا دو آرایه پویا برای دو رشته در نظر میگیرم. پس از دریافت دو رشته از ورودی یک آرایه پویا برای خروجی در نظر میگیرم و سپس با یک حلقه مقادیر را از s و t در r مقدار دهی می کنیم.

۵. کد زیر

```cpp
#include <iostream>

using namespace std;

void getNumbers(int* arr, int n) {
    for(int i = 0; i < n; i++) {
        cin >> arr[i];
    }
}

void sortNumbers(int* arr, int n) {
    cout << "Sorting the array ussing bubble method" << endl;
    for(int i = 0; i < n - 1; i++) {
        for(int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j+1]) {
                swap(arr[j], arr[j+1]);
            }
        }
    }
}

float findMedian(int* arr, int n) {
    if (n % 2 == 1) {
        return arr[n / 2];
    } else {
        return (float)(arr[n / 2] + arr[n / 2 - 1]) / 2;
    }
}

void showMedian(int* arr, int n) {
    float median = findMedian(arr, n);

    cout << "Median is: " << median << endl;
}

int main() {
    int n;
    cin >> n;

    int *arr = new int[n];

    getNumbers(arr, n);
    sortNumbers(arr, n);
    showMedian(arr, n);
}
```

تابع های ورودی و خروجی مشخص شده اند. قسمت اصلی تابع های مرتب سازی و پیدا کردن میانه می باشند. برای مرتب سازی از الگوریتم حبابی استفاده شده. برای پیدا کردن میانه هم با توجه به زوجیت n مقدار را از آرایه مرتب شده بدست می آوریم.