

جزوه مرور سریع

برنامه نویسی به زبان C++

تهیه شده توسط: امین نظری

نکته ۱: کدها را میتوانید از آدرس زیر دریافت نمایید.

<https://github.com/aminnazari91/BASU-CPP.git>

نکته ۲: این جزوه به مرور و در طول ترم تکمیل میشود.

نکته ۳: این کدها حداقل دانش مورد نیاز است که باید یاد گرفته باشید.

فصل صفرم:

آشنایی با متغیرها، عملگرها و دستورات ورودی/خروجی

مثال ۰: آشنایی با ساختار برنامه:

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

int main(){

    return 0;
}
```

مثال ۱: کار با خروجی

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

int main(){
    cout<<"Hello!"<<endl;
    return 0;
}
```

مثال ۲: تعریف متغیر

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

int main(){
    // variable declaration and assignment.
    int a, b, c;
    a = b = 2;
    c = a + b;
    cout<< a << " + " << b << " = " << c;
    cout<< &a << " " << &b << " " << &c << endl;

    return 0;
}
```

مثال ٣: انواع متغير

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

int main(){
    // variables types
    int int_number = 152;
    float pi = 3.14f;
    double d = 0.000000000001;
    cout<<"float: "<<pi<<"\ndouble: "<<d<<endl;

    char ch1 = 'a';
    char ch2 = 'A';
    cout<<"lowercase: "<<ch1 <<"\t uppercase: "<< ch2<<endl;

    string name="Ali";
    string family="Ahmadi";
    cout<< "Hello " <<name <<" " <<family<<endl;

    bool man = true;
    bool women = false;
    cout << "Ali is a man? " << man << " | " << !women <<endl;
    return 0;
}
```

مثال ٤: تفاوت float و double

```
// double and float precision values
// calculate roots of quadratic equation

#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    float fa = 1.0f;
    float fb = -4.0000000f;
    float fc = 3.9999999f;

    double da = 1.0;
    double db = -4.0000000;
```

```

double dc = 3.9999999;

double d = db * db - 4.0 * da * dc;
double sd = sqrt(d);
double r1 = (-db + sd) / (2.0 * da);
double r2 = (-db - sd) / (2.0 * da);

cout<<r1 <<"\t" <<r2<<endl;

float fd = fb * fb - 4.0f * fa * fc;
float fsd = sqrtf(fd);
float r3 = (-fb + fsd) / (2.0f * fa);
float r4 = (-fb - fsd) / (2.0f * fa);
cout<<r3 <<"\t" <<r4<<endl;
return 0;
}

```

مثال ۵: عملگرهای ریاضی

```

#include <iostream>
using namespace std;

int main(){
    int a, b;
    a = 2;
    b = 6;
    // arithmetic operators

    cout<< (a+b) << endl;
    cout<< (a-b) << endl;
    cout<< (a*b) << endl;
    cout<< (a/b) << endl;
    cout<< (a%b) << endl;
    cout<< (a++) << endl;
    cout<< a << endl;
    cout<< ++b << endl;

    return 0;
}

```

مثال ۶: عملگرهای مقایسه ای

```

#include <iostream>

```

```
using namespace std;

int main(){
    // comparison operators
    int a, b;
    a = 2;
    b = 6;

    cout<< (a>=b) << endl;
    cout<< (a>b) << endl;
    cout<< (a==b) << endl;
    cout<< (a!=b) << endl;
    cout<< (a<b) << endl;
    cout<< (a<=b) << endl;

    return 0;
}
```

مثال ۷: عملگرهای منطقی

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

int main(){
    // logical operators
    int a, b;
    a = 2;
    b = 6;

    cout<< ((a>=b) || (a<5)) << endl;
    cout<< ((a>b) && (a<5)) << endl;
    cout<< !(a==b) << endl;

    return 0;
}
```

مثال ۸: عملگرهای بیتی

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

int main(){
    // bitwise operators
```

```

int a, b;
a = 2;
b = 6;

cout<< (a<<2) << endl;
cout<< (b>>1) << endl;
cout<< (b&3) << endl;
cout<< (b|3) << endl;
return 0;
}

```

مثال ۹: عملیات ریاضی بر روی کاراکترها

```

#include <iostream>
using namespace std;

int main(){
    // character arithmetic oprantors

    char a, b, c, d;
    a = 'a';
    b = 'b';
    c = a + b;
    d = a * b;
    cout<< c << endl;
    cout<< d << endl;
    cout<< (a+b) << endl;
    cout<< (a*b) << endl;
    cout<< (a+1) << endl;
    cout<< (a-2) << endl;
    cout<< (a*2) << endl;
    cout<< (a/2) << endl;
    return 0;
}

```

مثال ۱۰: عملگر ریاضی روی رشته ها

```

#include <iostream>
using namespace std;

int main(){
    // character arithmetic oprantors

```

```

string name = "Ali";
string family = "Hasani";
string full_name = name + " " + family;
cout<< full_name<< endl;
return 0;
}

```

مثال ۱۱: short, long, unsigned

```

#include <iostream>
using namespace std;

int main(){

    // short or long

    int a = 5;
    long int b = 5;
    short int c = 5;
    unsigned int d = 5;
    unsigned long int e = 5;

    cout<< sizeof(a) << endl;
    cout<< sizeof(b) << endl;
    cout<< sizeof(c) << endl;
    cout<< sizeof(d) << endl;
    cout<< sizeof(e) << endl;

    d = -5;
    cout<<d<<endl;
    return 0;
}

```

مثال ۱۲: عملگر ؟ برنامه ای که ۳ صحیح را از ورودی گرفته و بزرگترین آنها را نمایش میدهد.

```

#include <iostream>
using namespace std;

int main(){
    int a, b, c, max;
    cout<<"Enter multiple numbers splited by spaces(a, b, c):";
    max = (a > b) ? ( (a > c) ? a : c ) : ( (b > c) ? b : c );
    cout << max;
    return 0;
}

```


مثال ۱۳: اعداد ثابت. برنامه ای که شعاع دایره را دریافت کرده و محیط و مساحت آنرا نمایش میدهد.

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

int main(){
    double r, s, p;
    const double pi = 3.14;
    cout<<"Enter a number:";
    cin >> r;
    s = pi * r * r;
    p = 2 * pi * r;

    cout<< "s=" << s << endl;
    cout<< "p=" << p<< endl;
    return 0;
}
```

اولویت عملگرها

Category	Operator	Associativity
Postfix	[] -> . ++ --	Left to right
Unary	+ - ! ~ ++ -- (type) * & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<< >>	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR		Left to right
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %= >>= <<= &= ^= =	Right to left
Comma	,	Left to right

مثال ۱۴: فرمت بندی ورودی و خروجی: <iomanip>

```
#include <iostream>
#include <iomanip>

using namespace std;

int main(){
    double d1, d2, d3;
    d1 = 3.14;
    d2 = 1.123456789;
    d3 = 2.0021;

    cout<< d1 << " " << d2 << " " << d3 <<endl;

    cout<< setw(12) << d1 << setw(12) << d2 << setw(12) << d3 <<endl;

    cout<< setw(12) << setprecision(2) << fixed << d2 <<endl;

    return 0;
}
```

تمرینات:

- ۱- برنامه ای بنویسید که طول و عرض یک مستطیل را گرفته و محیط و مساحت آنرا نمایش دهد.
- ۲- برنامه ای بنویسید که سه عدد را گرفته و میانگین آنرا نمایش دهد.
- ۳- برنامه ای بنویسید که دو عدد را گرفته و مقادیر آنها را جا به جا نماید.
- ۴- برنامه ای بنویسید که دما را به فارنهایت دریافت کرده و به سلسیوس نمایش دهد.
- ۵- برنامه ای که زاویه را برحسب درجه گرفته و به رادیان تبدیل نماید.
- ۶- کد اسکی یک کاراکتر را چاپ کنید.

فصل اول: ساختارهای کنترلی

مثال ١٥: if - قدر مطلق

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

int main(){
    int a;
    cout<<"Enter a number: ";
    cin >> a;
    // abs
    if (a<0){
        a = -a;
    }
    cout<< a << endl;
    return 0;
}
```

مثال ١٦: if-else - تعيين علامت

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

int main(){
    int a;
    cout<<"Enter a number: ";
    cin >> a;

    if (a<0){
        cout<<"Positive";
    }
    else{
        cout<<"Negative";
    }
    return 0;
}
```

مثال ۱۷: if-else if - محاسبه مالیات و خالص دریافتی

Salary < 200\$	معاف از مالیات
200\$<=Salary < 300\$	۵٪
300\$<=Salary < 400\$	۷٪
Salary >= 400\$	۱۰٪

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

int main(){
    float salary, net_salary, tax;
    cout<<"Enter your salary: ";
    cin >> salary;
    if (salary<=0){
        cout<<"Error";
    }
    else if (salary<200){
        tax = 0;
        net_salary = salary;
    }
    else if (salary<300){
        tax = (salary - 200) * 0.05;
        net_salary = salary - tax;
    }
}
```

```

    }
    else if (salary < 400){
        tax = 100 * 0.05 + (salary - 300) * 0.07;
        net_salary = salary - tax;
    }
    else{
        tax = 100 * 0.05 + 100 * 0.07 + (salary - 400) * 0.1;
        net_salary = salary - tax;
    }

    cout << "Tax: " << tax << endl << "Net Salary: " << net_salary;
    return 0;
}

```

مثال ۱۸: switch - روزهای هفته

```

#include <iostream>
using namespace std;

int main(){
    int day;
    cout << "Enter a number [1, 7]:";
    cin >> day;

    switch (day)
    {

        //, Friday, Saturday and Sunday
        case 1:
            cout << "Monday";
            break;

        case 2:
            cout << "Tuesday";
            break;

        case 3:
            cout << "Wednesday";
            break;

        case 4:
            cout << "Thursday";
            break;
    }
}

```

```

    case 5:
        cout << "Friday";
        break;

    case 6:
        cout << "Saturday";
        break;

    case 7:
        cout << "Sunday";
        break;

    default:
        break;
}
return 0;
}

```

مثال ۱۹: 1 switch-حروف صدا دار/ بی صدا

```

#include <iostream>
using namespace std;

int main(){
    char ch;
    cin>>ch;
    // convert to lowercase

    if (int(ch) >= int('A') && int(ch) <= int('Z'))
        ch = ch + 32;

    switch (ch){
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            cout<<" Vowels"; break;
        default:
            cout<<"Consonants";

    }

    return 0;
}

```

مثال ۲۰: switch-2 - تعداد ارقام یک عدد کوچکتر از ۱۰۰۰

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

int main(){
    int a;
    cin>>a;
    switch (a){
        case 0 ... 9:
            cout<<"1"; break;
        case 10 ... 99:
            cout<<"2"; break;
        case 100 ... 999:
            cout<<"3"; break;
        default:
            cout<<"Err";
    }
    return 0;
}
```

مثال ۲۱: for - چاپ اعداد ۱ تا ۱۰۰

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

int main(){
    for (int i=1; i<100; i++){
        cout << i <<"\t";
    }
    return 0;
}
```

مثال ۲۲: مجموع اعداد ۱ تا n

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

int main(){
    int n, sum = 0;
```



```

    cout<<"Enter a number: "<<endl;
    cin >> n;
    for (int i=1; i<=n; i++){
        sum += i;
    }
    cout<<sum;
    return 0;
}

```

مثال ۲۳: محاسبه فاکتوریل

```

#include <iostream>
using namespace std;

int main(){
    int n, fact = 1;
    cout<<"Enter a number: "<<endl;
    cin >> n;
    for (int i=1; i<=n; i++){
        fact *= i;
    }
    cout<<fact;
    return 0;
}

```

مثال ۲۴: while-مجموع ارقام یک عدد صحیح

```

#include <iostream>
using namespace std;

int main(){
    int n, sum = 0;
    cout<<"Enter a number: "<<endl;
    cin >> n;
    if (n<0)
        n = -n;

    while(n>0){
        int r = n % 10;
        sum += r;
        n = n/10;
    }
    cout<<sum;
    return 0;
}

```

مثال ۲۵: معکوس یک عدد صحیح

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

int main(){
    int n, reverse = 0;
    cout<<"Enter a number: "<<endl;
    cin >> n;

    int flag = 1;
    if (n<0){
        n = -n;
        flag = -1;
    }

    while(n>0){
        int r = n % 10;
        reverse = reverse * 10 + r;
        n = n/10;
    }
    cout<< flag * reverse;
    return 0;
}
```

مثال ۲۶: ترسیم اشکال زیر

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

int main(){
```

```

for (int i=1; i<=4; i++)
{
    for (int j=1; j<=8; j++)
        cout<<"*";
    cout<<endl;
}
return 0;
}

```

*

**

```

for (int i=1; i<=4; i++)
{
    for (int j=1; j<=i; j++)
        cout<<"*";
    cout<<endl;
}

```

*

**

```

for (int i=1; i<=4; i++)
{
    for (int j=1; j<=4-i; j++)
        cout<<" ";
    for (int j=1; j<=i; j++)
        cout<<"*";
    cout<<endl;
}

```

```

      *
     ***
    *****
   ********

```

```

for (int i=1; i<=4; i++)
{
    for (int j=1; j<=4-i; j++)
        cout<<" ";
    for (int j=1; j<=2*i-1; j++)
        cout<<"*";
    cout<<endl;
}

```

مثال ۲۷: n جمله از سری زیر را محاسبه کنید.

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

```

#include <iostream>
using namespace std;

int main(){
    float x, fact, pow, sin, cos;
    int sign, n = 10;
    cout<<"Enter a number(radian):";
    cin>>x;

    // sin(x)
    sin = 0;
    sign = 1;
    for (int i=1; i<=n; i++){
        pow = 1;
        for (int j=1; j<=2*i-1; j++){
            pow = pow * x;
        }

        fact = 1;
        for (int j=1; j<=2*i-1; j++){
            fact *= j;
        }

        sin += sign * (pow/fact);
    }
}

```

```

        sign = -sign;
    }

    //cos(x)
    cos = 1;
    sign = -1;
    for (int i=1; i<=n; i++){
        pow = 1;
        for (int j=1; j<=2*i; j++){
            pow *= x;
        }

        fact = 1;
        for (int j=1; j<=2*i; j++){
            fact *= j;
        }

        cos += sign * (pow/fact);
        sign = -sign;
    }

    cout<<"sin("<<x<<">sin<<endl;
    cout<<"cos("<<x<<">cos<<endl;
    return 0;
}

```

مثال ۲۸: اعداد رندوم <ctime> و <cstdlib>

```

#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main(){
    int t = time(0);
    cout<<t<<endl;
    srand(t);
    for (int i=0; i< 10; i++)
    {
        cout<<(rand()%1000)<<" ";
    }
    return 0;
}

```

مثال ۲۹: Break- برنامه ای که بی شمار عدد را دریافت و جمع میکند تا زمانی که عدد صفر وارد شود. سپس مجموع را نمایش میدهد.

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

int main(){
    int x, sum = 0;

    while(1){
        cout<<"Enter a number(for exit enter 0):"<<endl;
        cin>>x;
        if (x==0)
            break;
        sum += x;
    }
    cout<<sum;
    return 0;
}
```

مثال ۳۰: continue- برنامه ای که ۱۰ عدد را دریافت کرده و اگر عدد مثبت باشد آنها را با هم جمع میکنیم.

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

int main(){
    int i, x, sum = 0;
    i = 0;
    while(i++<10){
        cout<<"Enter a number:"<<endl;
        cin>>x;
        if (x<0)
            continue;
        sum += x;
    }
    cout<<sum;
```

```
    return 0;
}
```

مثال ۳۱: بازی حدس عدد.

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main(){
    int gole, user_guess, no_try = 0;

    srand(time(0));
    gole = rand()%1000 + 1;

    while(no_try++<10){
        cout<<"Enter your guess[1, 1000]:";
        cin >>user_guess;
        if (user_guess == gole){
            break;
        }

        else if (user_guess > gole){
            cout<<"Enter a lower number";
        }
        else{
            cout<<"Enter an upper number";
        }
    }
    if (no_try<=10){
        cout<<"You win!"<<"\nYour score is "<< (10-no_try) * 10<<" from 100";
    }
    else{
        cout<<"The number was"<<gole <<"You loss!";
    }
    return 0;
}
```

تمرینات:

۱- بازی سنگ، کاغذ، قیچی را بنویسید. بازی تا زمانی که یک طرف ۳ بار برنده شود ادامه پیدا میکند.
سپس امتیاز هر دو طرف چاپ میشود.

۲- برنامه ای بنویسید که n جمله از سری های زیر را محاسبه کند.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \quad \text{for } |x| < 1$$

$$\tan^{-1}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots \quad \text{for } |x| < 1$$

۳- مثلث خیام پاسکال را در خروجی نمایش دهید.

فصل دوم: توابع

انواع تابع:

تعداد ورودی	تعداد خروجی
-	-
1	-
N	-
1	1
N	1
-	1
-	M
1	M
N	M

مثال ۳۲: تابع بدون ورودی و خروجی

```

#include <iostream>
using namespace std;

void hello();

int main(){
    hello();
    return 0;
}

void hello(){
    string name;
    cout<<"Enter your name:";
    cin>>name;
    cout<<"Hello "<<name<<endl;
}

```

مثال ۳۳: تابع با یک ورودی و بدون خروجی

```

#include <iostream>
using namespace std;

void hello(string name);

int main(){
    string name;
    cout<<"Enter your name:";
    cin>>name;
    hello(name);
    return 0;
}

void hello(string name){
    cout<<"Hello "<<name<<endl;
}

```

مثال ۳۴: تابع با چند ورودی و بدون خروجی

```

#include <iostream>
using namespace std;

void add(int, int);

```

```

int main(){
    int a, b;
    a = 5;
    b = 6;
    add(a, b);
    return 0;
}

void add(int x, int y){
    cout<< (x + y) <<endl;
}

```

مثال ۲۵: تابع با یک ورودی و یک خروجی

```

#include <iostream>
using namespace std;

long int fact(int);

int main(){
    int x;
    long int f;
    cout<<"Enter a number:";
    cin>>x;
    f = fact(x);
    cout<<x<<"! = "<<f;
    return 0;
}

long int fact(int n){
    long int result = 1;
    for (int i=1; i<=n; i++){
        result *= i;
    }
    return result;
}

```

مثال ۲۶: تابع با چند ورودی و یک خروجی

```

#include <iostream>
using namespace std;

long int pow(int, int);

```

```

int main(){
    int base, power;
    long int result;
    base = 5;
    power = 3;
    result = pow(base, power);
    cout<<result;
    return 0;
}

long int pow(int x, int y){
    long int power = 1;
    for (int i=1; i<=y; i++){
        power *= x;
    }
    return power;
}

```

مثال ۳۷: تابع بدون ورودی و یک خروجی

```

#include <iostream>
using namespace std;

float Pi();

int main(){
    float r, S, P;
    cout<<"Enter the radius:";
    cin >>r;
    S = Pi() * r * r;
    P = 2 * Pi() * r;
    cout<<"S = " << S << " P = " << P;
    return 0;
}

float Pi(){
    return 3.14;
}

```

مثال ۳۸: حوزه متغیرها- خروجی کد زیر چیست؟

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

int main(){
    float j = 1.0f, result = 0;
    int i = 0;
    for(int i = 1; i<=5; i++, j *= 2)
        result += float(i/j);

    if (result > 2){
        float j = 2;
    }
    cout << result << endl << i << endl << j;
}
```

مثال ۳۹: متغیر سراسری و متغیر محلی

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

int s=10;
void func1(){
    int s = 0;
    s++;
    cout<< "Local variable:" << s <<endl;
}

void func2(){
    s++;
    cout<< "global variable:" << s <<endl;
}

int main(){
    func1();
    func1();
    func2();
    func2();
    cout<<"main " << s <<endl;
    return 0;
}
```

مثال ۴۰: متغیر استاتیک

```
#include <iostream>
```

```
using namespace std;

int s=10;
void func(){
    static int s = 0;
    s++;
    cout<< "Local static: "<< s <<endl;
}
int main(){
    func();
    func();
    func();
    func();

    cout<< "Global:" << s <<endl;
    return 0;
}
```

مثال ۴۱: مقادیر پیش فرض ۱

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

int add(int x=1, int y=2){
    return (x+y);
}
int main(){
    cout<< "Use defaults values:" << add() <<endl;
    cout<< "Use default value for the first parameters:" << add(5) <<endl;
    cout<< "Don't use default values:" << add(5, 12) <<endl;
    return 0;
}
```

مثال ۴۲: مقادیر پیش فرض ۲ (از سمت راستترین پارامتر باید مقدار پیش فرض ست شود.)

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

void print_message_to(string name, string msg="Hello" ){
    cout << msg + " " + name <<endl;
}
int main(){
```

```

    print_message_to("Ali");
    print_message_to("Hasan", "Wellcome");
    return 0;
}

```

مثال ۴۲: توابع همنام

```

#include <iostream>
using namespace std;

int add(int x, int y){
    cout <<"int_add: ";
    return (x+y);
}

float add(float x, float y){
    cout <<"float_add: ";
    return (x+y);
}

string add(string x, string y){
    cout <<"string_add: ";
    return (x+y);
}

int main(){
    cout<< add(4, 5) << endl;
    cout<< add(4.0, 5) << endl;
    cout << add(4.0f, 5.0f) << endl;
    cout << add("4", "5") << endl;

    return 0;
}

```

تمرین:

۱- توابع زیر را بنویسید.

فاکتوریل، فیبوناچی، توان، قدرمطلق، e, cot, tan, cos, Sin

فصل سوم:

آرایه ها

مثال ۴۳: تعریف آرایه، خواندن عناصر از ورودی و ذخیره در آرایه و نمایش دو برابر عناصر

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

int main(){
    const int n = 5;
    int A[n];

    for (int i=0; i<n; i++){
        cout<<"Enter the "<< i+1 << "'th element:";
        cin>> A[i];
    }

    for (int i=0; i<n; i++){
        cout<< 2 * A[i]<<"\t";
    }
    return 0;
}
```

مثال ۴۴: مقدار اولیه به آرایه

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

int main(){
    float B[] = {1.0, 2.3, 2.5, 5.1};
    int sizeB = sizeof(B) / sizeof(B[0]);

    for (int i=0; i<sizeB; i++){
        cout<<B[i]<<"\t";
    }
    return 0;
}
```

مثال ۴۵: تعیین طول آرایه در زمان اجرا

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

int main(){
    int n;
```

```

cout<<"Enter the length of the array:";
cin>>n;
int A[n];

for (int i=0; i<n; i++){
    cout<<"Enter the "<< i+1 << "'th element:";
    cin>> A[i];
}

for (int i=0; i<n; i++){
    cout<< 2 * A[i]<<"\t";
}
return 0;
}

```

مثال ۴۶: آرایه به عنوان پارامتر تابع - نکته: تغییرات آرایه در تابع، توسط تابع اصلی قابل مشاهده است.
(برخلاف متغیر محلی)

```

#include <iostream>
using namespace std;

void input_array(int X[], int len){
    for(int i=0; i<len; i++){
        cout<<"Enter the "<<i+1<<"'th element:";
        cin>>X[i];
    }
}

void print_array(int X[], int len){
    for(int i=0; i<len; i++){
        cout<<X[i]<<"\t";
    }
}

int main(){
    const int n = 5;
    int A[n];

    input_array(A, n);
    print_array(A, n);

    return 0;
}

```

مثال ۴۷: مرتب سازی آرایه: مرتب سازی دودویی $n-1$ مرحله انجام میشود که در هر مرحله از ابتدا تا عنصر $n-1$ ام بررسی میشود. اگر عنصر با عنصر بعدی خود بررسی میشود و در صورت نیاز با آن جا به جا میشود.

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

void input_array(int X[], int len){
    for(int i=0; i<len; i++){
        cout<<"Enter the "<<i+1<<"\'th element:";
        cin>>X[i];
    }
}

void print_array(int X[], int len){
    for(int i=0; i<len; i++){
        cout<<X[i]<<"\t";
    }
}

void sort(int X[], int len){
    for (int i=0; i<len-1; i++){
        for(int j=0; j<len-1; j++){
            if (X[j]> X[j+1]){
                int temp;
                temp = X[j];
                X[j] = X[j+1];
                X[j+1] = temp;
            }
        }
    }
}

int main(){
    const int n = 5;
    int A[n];

    input_array(A, n);
    sort(A, n);
    print_array(A, n);

    return 0;
}
```

مثال ۴۸ : دسترسی به عناصر خارج از طول آرایه- (*)

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

int main(){
    int A[] = {2,2,4,6,7};
    for(int i=0; i<5; i++){
        cout<<i<<"\t";
    }
    return 0;
}
```

مثال ۴۹ : اشتراک دو آرایه:

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

int main(){

    int A[] = {1,1,2,3,3,4};
    int B[] = {2,2,4,4,6,6};

    int sizeA = sizeof(A) / sizeof(A[0]);
    int sizeB = sizeof(A) / sizeof(A[0]);
    int sizeC = sizeA<sizeB ? sizeA: sizeB;
    int C[sizeC];

    int i, j, k = 0;
    for (i=0; i<sizeA; i++){
        for(j=0; j<sizeB; j++){
            if(A[i] == B[j]){
                break;
            }
        }

        if (j == sizeB){
            continue;
        }

        for(j=0; j<k; j++){
            if(A[i] == C[j]){
                break;
            }
        }
    }
}
```

```

    }
    if (j == k){
        C[k++] = A[i];
    }
}

for (i=0; i<k; i++){
    cout<<C[i] <<"\t";
}
return 0;
}

```

مثال ۵۰: اجتماع دو آرایه:

```

#include <iostream>

using namespace std;

int main(){

    int A[] = {1,1,2,3,3,4};
    int B[] = {2,2,4,4,6,6};

    int sizeA = sizeof(A) / sizeof(A[0]);
    int sizeB = sizeof(A) / sizeof(A[0]);

    int C[sizeA + sizeB];

    int i, j, k = 0;
    for (i=0; i<sizeA; i++){
        for(j=0; j<k; j++){
            if(A[i] == C[j]){
                break;
            }
        }

        if (j == k){
            C[k++] = A[i];
        }
    }

    for (i=0; i<sizeB; i++){
        for(j=0; j<k; j++){

```

```

        if(B[i] == C[j]){
            break;
        }
    }

    if (j == k){
        C[k++] = B[i];
    }

}

for (i=0; i<k; i++){
    cout<<C[i] <<"\t";
}

return 0;
}

```

مثال ۵۱: تفاضل دو آرایه

```

#include <iostream>
using namespace std;

int main(){

    int A[] = {1,1,2,3,3,4};
    int B[] = {2,2,4,4,6,6};

    int sizeA = sizeof(A) / sizeof(A[0]);
    int sizeB = sizeof(B) / sizeof(B[0]);
    int sizeC = sizeA<sizeB ? sizeA: sizeB;
    int C[sizeC];

    int i, j, k = 0;
    for (i=0; i<sizeA; i++){
        for(j=0; j<sizeB; j++){
            if(A[i] == B[j]){
                break;
            }
        }

        if (j == sizeB){
            for(j=0; j<k; j++){
                if(A[i] == C[j]){
                    break;
                }
            }
        }
    }
}

```

```

    }
}

    if (j == k){
        C[k++] = A[i];
    }
}

for (i=0; i<k; i++){
    cout<<C[i] <<"\t";
}
return 0;
}

```

مثال ۵۲: تابع جستجو- تابع ای که یک آرایه و یک عدد را دریافت کرده و اگر آن عدد در آرایه باشد، اندیس عدد و در غیر این صورت منفی یک برمیگرداند.

```

#include <iostream>
using namespace std;
int find(int X[], int len, int item);
int main(){
    int A[]={1,2,3,4,5};
    int idx = find(A, 5, 2);
    cout<<idx;
    return 0;
}

int find(int X[], int len, int item){
    for (int i=0; i<len; i++){
        if (X[i] == item){
            return i;
        }
    }
    return -1;
}

```

مثال ۵۳: جستجوی دودویی (با فرض آنکه آرایه مرتب باشد، برای جستجوی یک عدد، آنرا با عنصر وسط آرایه

مقایسه میکنیم. اگر عنصر برابر بود، اندیس را برمیگردانیم. اگر عنصر وسط بزرگتر بود باید عدد را در بخش سمت چپ جستجو کنیم. اگر عنصر وسط کوچکتر بود، باید عدد را در سمت راست آرایه جستجو نماییم.)

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

void input_array(int X[], int len){
    for(int i=0; i<len; i++){
        cout<<"Enter the "<<i+1<<"\th element:";
        cin>>X[i];
    }
}

int binary_search(int X[], int len, int item){
    int high, low, mid;

    high = len-1;
    low = 0;
    while(high>=low){
        mid = (high+low) / 2;
        if (X[mid] == item)
            return mid;
        else if(X[mid] > item)
            high = mid - 1;
        else
            low = mid + 1;
    }
    return -1;
}

void sort(int X[], int len){
    for (int i=0; i<len-1; i++){
        for(int j=0; j<len-1; j++){
            if (X[j]> X[j+1]){
                int temp;
                temp = X[j];
                X[j] = X[j+1];
                X[j+1] = temp;
            }
        }
    }
}

int main(){
```



```

const int n = 5;
int A[n] = {1,2,3,4,5};

sort(A, n);
for (int i=0; i<10; i++){
    int idx = binary_search(A, n, i);
    cout<<i <<" = " <<idx<<endl;
}
return 0;
}

```

مثال ۵۴: مرتب سازی در زمان دریافت داده ها

```

#include <iostream>
using namespace std;
int main(){

    const int n = 5;
    int A[n];

    int i, j;
    for(i=0; i<n;i++){
        {
            int temp;
            cin>> temp;
            j = 0;
            while(j<i){
                if (A[j]>temp)
                    break;
                j++;
            }
            if(j==i){
                A[j] = temp;
            }
            else{
                for(int k=i; k>=j; k--){
                    A[k] = A[k-1];
                }
                A[j] = temp;
            }
        }
    }

    for(i=0;i<n; i++)
        cout<<A[i]<<"\t";
}

```

```
    return 0;
}
```

مثال ۵۵: آرایه های دو بعدی

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

int main(){
    int n = 5;
    int m = 2;

    cin >>n >> m;
    int A[n][m];
    for(int i=0; i<n; i++){
        for (int j=0; j<m; j++){
            cin>>A[i][j];
        }

        for(int i=0; i<n; i++){
            for (int j=0; j<m; j++){
                cout<<A[i][j] <<"\t";
            }
            cout<<endl;
        }
    }
}
```

مثال ۵۶ تا ۶۲: تابعی برای جمع دو ماتریس / تابعی برای ضرب دو ماتریس / تابعی برای بررسی تقارن دو ماتریس / تابعی برای ترانزهاذ کردن یک ماتریس، بررسی تساوی دو ماتریس، تابعی که چک کند که آیا ماتریس یک ماتریس بالامثلثی یا پایین مثلثی است. ماتریسی ماتریس پایین مثلثی است که تمام درایه های بالای قطر اصلی آن صفر باشد. به همین ترتیب، ماتریسی که تمام درایه های زیر قطر اصلی آن صفر است، یک ماتریس بالا مثلثی است.

```
#include <iostream>
using namespace std;
void Matrix_Addition(int A[][2], int B[][2], int C[][2], int rows){
    for (int i=0;i<rows;i++)
        for (int j=0;j<2; j++)
            C[i][j] = A[i][j] + B[i][j];
}
```

```

}
void Zeros(int A[2][2]){
    for (int i=0;i<2;i++)
        for (int j=0;j<=i; j++){
            A[i][j] = 0;
            A[j][i] = 0;
        }
}

void Matrix_Multiplication(int A[2][2], int B[2][2], int C[2][2]){
    for (int i=0;i<2;i++)
        for (int j=0;j<2; j++)
            for (int k=0; k<2; k++)
                C[i][j] += A[i][k] * B[k][j];
}

bool Is_Symmetrical(int A[][2], int rows){
    for (int i=0;i<rows;i++)
        for (int j=0;j<2; j++)
            if (A[i][j] != A[j][i])
                return false;

    return true;
}

void Transpose(int A[2][2]){
    for (int i=0;i<2;i++)
        for (int j=0;j<i; j++){
            int temp = A[i][j];
            A[i][j] = A[j][i];
            A[j][i] = temp;
        }
}

bool Is_Equal(int A[2][2], int B[2][2]){
    for (int i=0;i<2;i++)
        for (int j=0;j<i; j++)
            if(A[i][j] != B[i][j])
                return false;

    return true;
}

bool Is_Upper_Triangular(int A[2][2]){
    for (int i=0;i<2;i++)
        for (int j=0;j<i; j++)
            if(A[i][j] != 0)

```

```

        return false;

    return true;
}

bool Is_Lower_Triangular(int A[2][2]){
    for (int i=0;i<2;i++){
        for (int j=1;j>i; j--){
            if(A[i][j] != 0)
                return false;
        }

        return true;
    }
}

void Print(int A[2][2]){
    for (int i=0;i<2;i++){
        for (int j=0;j<2; j++){
            cout<<A[i][j]<<"\t";
            cout<<endl;
        }
        cout<<endl;
    }
}

int main(){
    int A[][2] = {{1, 2}, {3, 4}};
    int B[][2] = {1, 0, 8, 9};
    int rows = 2;
    int C[rows][2];

    cout<<"A: "<<endl;
    Print(A);

    cout<<"B: "<<endl;
    Print(B);

    cout<<"C: "<<endl;
    Print(C);

    Zeros(C);
    cout<<"C: "<<endl;
    Print(C);

    Matrix_Addition(A, B, C, rows);
    cout<<"A+B:"<<endl;
    Print(C);
}

```

```

Zeros(C);
Matrix_Multiplication(A, B, C);
cout<<"AxB:"<<endl;
Print(C);

cout<<"B: "<<endl;
Print(B);
cout<<"Is_Lower_Triangular:"<< Is_Lower_Triangular(B) <<endl;
cout<<"Is_Upper_Triangular:"<< Is_Upper_Triangular(B) <<endl;

Transpose(B);

cout<<"B': "<<endl;
Print(B);
cout<<"Is_Lower_Triangular:"<< Is_Lower_Triangular(B) <<endl;
cout<<"Is_Upper_Triangular:"<< Is_Upper_Triangular(B) <<endl;

cout<<"B' "<<endl;
Print(B);
cout<<"Is_Symmetrical:"<< Is_Symmetrical(B, 2) <<endl;

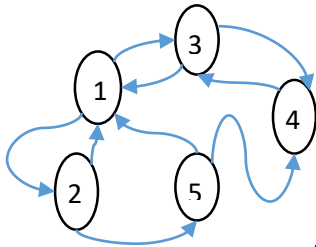
cout<<"A == A"<<endl;
Print(A);
cout<<"Is_Equal:"<< Is_Equal(A, A) <<endl;
}

```

تمرین:

- ۱- با استفاده از آرایه ها، فاکتوریل و فیبوناچی را محاسبه کنید.
- ۲- با استفاده از ماتریس مسئله انتخاب را حل کنید. $\binom{n}{m}$
- ۳- شکل زیر جاده های ارتباطی بین شهرهای مختلف را نشان میدهد. یک مهندس راه قرار است جاده های جدیدی بین این شهرها احداث کند که راه های یکطرفه را به جاده های دوطرفه تبدیل نماید. شرط زیر برای احداث جاده های جدید برقرار است.

۴- اگر جاده ی رفت با برگشت برابر باشد، جاده جدید یک دهم جاده ی قدیمی خواهد بود در غیر این صورت جاده احدائی برابر با میانگین جاده های یکطرفه است.



الف) تابعی بنویسید که اطلاعات راه ها را دریافت کرده و در ساختار مناسبی ذخیره نماید.

ب) تابعی بنویسید که با دریافت طول راه های قدیمی (با کمک بخش الف)، طول راه های جدید را محاسبه نماید.

۵- تابعی بنویسید که دو آرایه از کارکترها را دریافت کرده و مشخص کنید که آیا آرایه دوم در آرایه اول وجود دارد یا خیر.

فصل چهارم:

اشاره گرها

مثال ۶۳: اشاره گرها (در اشاره گر، آدرس (بایت اول) یک متغیر ذخیره میشود) آدرس متغیر با & قابل در دسترسی است. بنابراین برای یک متغیر دو راه دسترسی خواهیم داشت. اولی از طریق نام متغیر و دومی از طریق اشاره گری که به آن متغیر اشاره میکند.

```
#include <iostream>
#include <cstring>
using namespace std;
void circle(float r, float *, float *);

int main(){
    int n = 10, *p;
    p = &n;
    cout << n << "\t" << &n << endl;
    cout << p << "\t" << &p << "\t" << *p << endl;
    *p += 10;
    cout << n << endl;

    n -= 2;
    cout << *p << endl;
    return 0;
}
```

در مثال بالا p آدرس متغیر n را نگه میدارد. برای دسترسی به محتوای متغیر از *p استفاده میکنیم.

مثال ۶۴: توابع با بیش از یک خروجی

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

```

void circle(float r, float *, float *);

int main(){

    float r, x, y;
    r = 2.5;
    circle(r, &x, &y);
    cout<<"S = "<<x<<"P = "<<y;
    return 0;
}

void circle(float r, float *S, float *P){
    *S = 3.14 * r * r;
    *P = 2 * 3.14 * r;
}

```

مثال ۶۵: آرایه ها و اشاره گرها (اسم تابع مانند یک اشاره گر است. برای همین در هنگام ارسال آرایه به تابع هر تغییری که در تابع محلی بر روی آرایه انجام میشود، در تابع اصلی حفظ میگردد)

```

#include <iostream>
#include <cstring>
using namespace std;

int main(){

    char A[]="This is a test";
    int len = sizeof(A) / sizeof(A[0]);

    char *p;
    p = A;

    for(int i=0; i<len; i++){
        cout<< *(p+i) <<endl;
    }
    return 0;
}

```

در کد بالا، p به خانه اول آرایه (اندیس صفرم) اشاره میکند برای دسترسی به خانه های بعدی کافیست تا ادرس را با شماره اندیس جمع کنیم. بنابراین $*(p+i)$ برابر با محتوای خانه $A[i]$ است.

مثال ۶۶: الگوها

```

#include <iostream>

```



```
using namespace std;

template <class T1>
T1 add(T1 a, T1 b){
    return a+b;
}

int main(){
    int a = 5, b = 6, c;
    float x = 5.5, y = 6.3, z;
    c = add<int>(a, b);
    cout<<c<<"\n";
    z = add<float>(x, y);
    cout<<z<<"\n";
    cout<<add<char>('A', 'C')<<"\n";
    return 0;
}
```

با استفاده از الگو میتوان یک نوع عمومی تعریف کرد و در زمان فراخوانی نوع متغیر را مشخص نمود. در مثال بالا تابع add برای انواع مختلف قابل اجراست.

مثال ۶۷: ارجاع به یک متغیر

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

int main(){
    int a = 10;
    int &b = a;

    b = 4;
    cout<<a<<"", "<<b;
    return 0;
}
```

ارجاع مانند نام مستعار عمل میکند. دو اسم برای یک متغیر داریم.

مثال ۶۸: ارجاع به عنوان متغیر بازگشتی از تابع.

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

```
int& max(int& m, int& n)
{ return (m > n ? m : n);}

int main(){
    int m = 44, n = 22;
    cout << m << ", " << n << ", " << max(m, n) << endl;

    max(m, n) = 55;

    cout << m << ", " << n << ", " << max(m, n) << endl;
}
```

در مثال بالا تابع max ادرس متغیر بزرگتر را به عنوان خروجی برمیگرداند. بنابراین میتوانیم یک مقدار را به تابع نسبت دهیم. در کد بالا، خروجی تابع max اسم متغیری است که بزرگترین مقدار را دارد. سپس ما به آن متغیر مقدار ۵۵ را نسبت داده ایم. سپس همه متغیرها را چاپ میکنیم.

مثال ۶۹: اشاره گر به تابع

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

int add(int a, int b){
    return a+b;
}

int main(){
    int (*p)(int, int);
    p = add;
    int s = p(5,6);
    cout<<s;
    return 0;
}
```

مثال ۷۰: اشاره گر به اشاره گر:

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

int main(){

    int x=5;
```

```

int *p, **q;

p = &x;
q = &p;

cout<<&p<<" = "<<q<<endl;

cout<<*p<<" = "<<**q<<endl;

**q = 6;
cout<<x<<endl;

return 0;
}

```

مثال ۷۱: اشاره گر و ماتریس

```

#include <iostream>
using namespace std;

void set_mat(int *a, int n){
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            if (i == j)
                *(a+i*n+j) = 0;
            else{
                cout<<"Enter edge ("<<i+1<<","<<j+1<<"): ";
                cin>> *(a+i*n+j);
            }
}

void get_mat(int *a, int n){
    for (int i=0; i<n; i++){
        for (int j=0; j<n; j++)
            cout<< *(a+i*n+j)<<"\t";
        cout<<"\n";
    }
}

void get_mat(double *a, int n){
    for (int i=0; i<n; i++){
        for (int j=0; j<n; j++)
            cout<< *(a+i*n+j)<<"\t";
        cout<<"\n";
    }
}

```

```

    }
}

void new_path(int *a, double *b, int n){

    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            *(b+i*n+j) = *(a+i*n+j);

    for (int i=1; i<n; i++){
        for (int j=0; j<i; j++){
            if ( *(a+i*n+j) == *(a+j*n+i)){
                *(b+i*n+j) -= *(a+i*n+j) * 0.1;
                *(b+j*n+i) = *(b+i*n+j);
            }
            else{
                *(b+i*n+j) = (*(a+i*n+j) + *(a+j*n+i)) / 2;
                *(b+j*n+i) = *(b+i*n+j) ;
            }
        }
    }
}

int main(){
    int n, *a;
    double *b;
    cout<<"Enter No cities:";
    cin>>n;

    a = new int [n*n];
    b = new double [n*n];

    set_mat(a, n);
    get_mat(a, n);
    new_path(a, b, n);
    get_mat(b, n);
    return 0;
}

```

فصل پنجم:

ساختارها

مثال ۷۲: struct

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

struct point{
    float x=0, y=0;
};

float distance(point p1, point p2){
    return sqrt(pow((p1.x-p2.x), 2) + pow((p1.y-p2.y), 2));
}

int main(){
    point p,q;
    cout<<"("<<p.x<<" "<<p.y<<"")<<setw(20);
    q.x = 10;
    q.y = 10;
    cout<<"("<<q.x<<" "<<q.y<<"")<<endl;
    cout<<"distance: "<<distance(p,q);
    return 0;
}
```

مثال ۷۳: آرایه ای از struct

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

struct student{
    string name;
    int age;
    float avg;
};

void print_std(student *p, int n){
    for (int i=0; i<n; i++)
        cout<<"("<<p[i].name<<" "<<p[i].age<<" "<<p[i].avg<<"")<<endl;
}

int main(){
    int n;
```

```

cout<<"Enter number of student";
cin>>n;
student s[n];
if (!s){
    cout<<"Error";
    return 1;
}
for (int i=0;i<n;i++){
    cout<<"name:";
    cin>>s[i].name;

    cout<<"age:";
    cin>>s[i].age ;

    cout<<"avg:";
    cin>>s[i].avg;
}
print_std(s, n);

return 0;
}

```

مثال ٧٤: توابع عضو ساختمان

```

#include <iostream>
using namespace std;
struct point{
    int x=0;
    int y=0;
    void set_point(int a, int b){

        this->x = a;
        this->y = b;
        //return this;
    }
    void get_point(){
        cout<<this->x<<"\t"<<this->y;
    }
};

struct line {
    point p1, p2;
};

```

```

int main(){
    line l1;
    point p;
    p.set_point(12, 5);
    p.get_point();
    /*
    cout<<"x:"<<l1.p1.x<<"\t"<<l1.p2.x<<endl;
    cout<<"y:"<<l1.p1.y<<"\t"<<l1.p2.y;
    */
    return 0;
}

```

مثال ۷۵: استفاده از struct در struct

```

#include <iostream>
#include <iomanip>
#include <cmath>
#include <string>
using namespace std;

struct point{
    int x=0, y=0;
    string name;
    void set(int a, int b){
        this->x = a;
        this->y = b;
    }
    void get(){
        cout<<"("<<this->x<<","<<this->y<<")"<<endl;
    }
};

struct line{
    point S,E;
    void set(point p1, point p2){
        this->S = p1;
        this->E = p2;
    }
    void set(int a, int b, int c, int d){
        point p1, p2;
        p1.set(a,b);
        p2.set(c,d);

        this->S = p1;

```



```

        this->E = p2;
    }
    void get(){
        cout<<"Start point:";
        this->S.get();
        cout<<"End point:";
        this->E.get();
    }
};
//p.set(5,6)// p.x = 5, p.y = 6
point add_point(point a, point b){
    point c;
    c.x = a.x + b.x;
    c.y = a.y + b.y;
    return c;
}
void set_points(point *p, int n){
    for (int i=0; i<n; i++){
        cout<<"Enter point "<< i +1 <<".x :";
        cin>>*(p+i).x;
        cout<<"Enter point "<< i +1 <<".y :";
        cin>>p[i].y;
    }
}
double distance(point a, point b){
    return sqrt(pow((a.x-b.x), 2) + pow((a.y-b.y), 2));
}
int main(){

    /*
    point p1,p2,p3,p4;
    p1.x = 10;
    p1.y = 20;
    p2.x = 100;
    p2.y = 200;

    p3 = add_point(p1, p2);
    cout<<"p1"<<endl;
    cout<<(" "<<p1.x<<","<<p1.y<<")"<<endl;

    cout<<"p2"<<endl;
    cout<<(" "<<p2.x<<","<<p2.y<<")"<<endl;

    cout<<"p3"<<endl;
    cout<<(" "<<p3.x<<","<<p3.y<<")"<<endl;

```

```

cout<<"p4"<<endl;
cout<<"("<<p4.x<<","<<p4.y<<")"<<endl;
*/
/*
int n;
point *p;
cout<<"Enter No Points:";
cin>>n;
p = new point [n];
set_points(p, n);
*/
/*
point p1, p2;
p2.set(5,6);
cout<<"p1"<<endl;
//cout<<"("<<p1.x<<","<<p1.y<<")"<<endl;
p1.get();

cout<<"p2"<<endl;
//cout<<"("<<p2.x<<","<<p2.y<<")"<<endl;
p2.get();

//cout<<distance(p1, p2);
*/
/*
point p1, p2;
p1.set(5,5);
line l1;
l1.set(p1, p2);
l1.get();
*/
int x=0,y=5,z=10,w=60;
line l1;
l1.set(x, y, z, w );

l1.get();
return 0;
}

```

مثال ۷۶: پیاده سازی صف به صورت لیست پیوندی

فصل ششم:

فایلها

مثال ۷۸: فایل‌های متنی و دسترسی ترتیبی - با فاصله هر بخش جدا میشود.

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <stdlib.h>
using namespace std;

void write_string_in_file(){
    ofstream of;
    of.open("f2.txt", ios::app);
    if(!of){
        cout<<"Error";
        exit(1);
    }
    string name;
    cout<<"Enter name:";
    cin>>name;

    int age;
    cout<<"Enter age:";
    cin>>age;

    float avg;
    cout<<"Enter avg:";
    cin>>avg;

    of<<name<<" "<<age<<" "<<avg<<"\n";

    of.close();
}

void read_line_in_file(){
    ifstream in;
    in.open("f2.txt");
    if(!in){
        cout<<"Error";
        exit(1);
    }

    string line;
    while(getline(in, line)){
        cout<<line<<endl;
    }
}
```

```

        in.close();
    }

void write_char_in_file(){
    ofstream out;
    out.open("f1.txt", ios::app);
    if(!out){
        cout<<"Error in open file!";
        exit(1);
    }
    char ch;
    cin.get(ch);
    while(ch != '\n'){
        out.put(ch);
        cin.get(ch);
    }
    out.put('\n');
    out.close();
}

void read_char_from_file(){
    ifstream in;
    in.open("f1.txt", ios::in);
    if(!in){
        cout<<"Errn";
        exit(1);
    }

    while(!in.eof()){
        char ch;
        in.get(ch);
        cout<<ch;
    }
    in.close();
}

void search(){
    ifstream in;
    in.open("f2.txt");
    if(!in){
        cout<<"Error";
        exit(1);
    }
}

```

```

    char name[21];
    int age;
    float avg;
    char keyword[21];
    cin.ignore();
    cout<<"Enter key:";
    cin.get(keyword, 21);
    while(in>>name>>age>>avg){
        if (strcmp(name, keyword)==0)
            cout<<name<<": "<<avg;
    }
    in.close();
}

int main(){
    system("color 0A");

    cout<<"Enter your txt then press \"Enter\":";
    write_char_in_file();
    cout<<"Display all content of the file(char_by_char):"<<endl;
    read_char_from_file();
    cout<<endl<<"Add a user in file:"<<endl;
    write_string_in_file();
    cout<<endl<<"Add a user in file:"<<endl;
    write_string_in_file();
    cout<<"Display all content of the file(line_by_line):"<<endl;
    read_line_in_file();
    cout<<endl<<"search in file:"<<endl;
    search();
    return 0;
}

```

مثال ۷۹: فایل متنی - هر بخش با | جدا میشود و میتوانیم متنیهایی که شامل فاصله هم هستند داشته باشیم.

```

#include <iostream>
#include <fstream> //file header
#include <string>
#include <bits/stdc++.h>

using namespace std;

struct student{

```

```

    string name;
    float score;
};

void set_array(student st[], int len){
    //get information 3 time
    for (int i = 0; i < len; i++){
        cout << "Enter Name: ";
        getline(cin, st[i].name);

        cout << "Enter score: ";
        cin >> st[i].score;
        cout << endl;

        cin.ignore();
    }
}

void write_file(student st[], int len){

    // define write file stream
    ofstream outputFile("output.dat", ios::out);

    //check for file opening
    if ( !outputFile){
        cerr << "some thing wrong during opening file!" << endl;
        exit(1);
    }

    for (int i = 0; i < len; i++){
        //write in file
        outputFile << st[i].name << "|" << st[i].score << endl;
    }
}

void read_file(student st[], int len){

    // define write file stream
    ifstream inputFile("output.dat", ios::in);

    //check for file opening
    if ( !inputFile) {
        cerr << "some thing wrong during opening file!" << endl;
    }
}

```

```

        exit(1);
    }
    // Use a while loop together with the getline() function to read the file
line by line
    string line;
    while (getline(inputFile, line)) {
        // Output the text from the file
        cout << line << endl;
    }
}

float avg_scores(){
    // define write file stream
    ifstream inputFile("output.dat", ios::in);

    //check for file opening
    if ( !inputFile) {
        cerr << "some thing wrong during opening file!" << endl;
        exit(1);
    }

    // Use a while loop together with the getline() function to read the file line
by line
    string line;
    float sum = 0.0f;
    float count_line = 0.0f;
    while (getline(inputFile, line)) {
        stringstream str(line);
        string part;
        int i = 0;

        while (getline(str, part, '|')){

            if (i == 1){
                char * tab2 = new char [part.length()+1];
                strcpy (tab2, part.c_str());
                cout<<tab2<<endl;
                sum += atof(tab2);
                cout<<sum<<endl;
            }
            i++;
        }
        count_line++;
    }
}

```



```

    return sum/count_line;
}
int main()
{
    cout<<avg_scores();

    return 0;
}

```

مثال ۸۰: فایل باینری و دسترسی تصادفی:

```

#include <iostream>
#include <fstream>

using namespace std;

struct student{
    int id;
    char name[21];
    int age;
    float avg;
}st;

void read_binary(){
    fstream file;
    file.open("f3", ios::in|ios::binary);

    while (file.read((char *) &st, sizeof(st))){
        cout<<st.name<<" "<<st.avg<<" "<<st.age<<endl;
    }
    file.close();
}

void search(){
    fstream file;
    file.open("f3", ios::in|ios::binary);
    int key;
    cin>>key;
    file.seekg(key*sizeof(st), ios::beg);
    file.read((char *) &st, sizeof(st));
    cout<<st.name<<" "<<st.avg<<" "<<st.age<<endl;

    file.close();
}

```

```

void edit(){
    fstream file;
    file.open("f3", ios::in|ios::out|ios::binary);
    int key;
    cin>>key;
    file.seekg(key*sizeof(st), ios::beg);
    file.read((char *) &st, sizeof(st));
    cout<<st.name<<" "<<st.avg<<" "<<st.age<<endl;
    cout<<"Enter name: ";
        cin>>st.name;

        cout<<"Enter age: ";
        cin>>st.age;

        cout<<"Enter avg: ";
        cin>>st.avg;
        file.seekp(st.id*sizeof(st), ios::beg);
        file.write((char *) &st, sizeof(st));
    file.close();
}

```

```

void edit2(){
    fstream in;
    in.open("f3", ios::in|ios::binary);

    fstream out;
    out.open("temp", ios::out|ios::binary);
    int c=0;
    while (in.read((char *) &st, sizeof(st))){
        if (st.id!=0){
            //out.seekp(c*sizeof(st), ios::beg);
            out.write((char *) &st, sizeof(st));
            cout<<st.id<<endl;
            //c++;
        }
    }

    in.close();

    out.close();
    remove("f3");
    rename("temp", "f3");
}

```

```

int main(){

    fstream file;
    file.open("f3", ios::out|ios::binary);
    while(1){
        char ch;
        cout<<"Do you want add another user. y/n: ";
        cin>>ch;
        //cin.ignore();
        if (ch != 'y')
            break;
        cout<<"Enter id: ";
        cin>>st.id;

        cout<<"Enter name: ";
        cin>>st.name;

        cout<<"Enter age: ";
        cin>>st.age;

        cout<<"Enter avg: ";
        cin>>st.avg;
        file.seekp(st.id*sizeof(st), ios::beg);
        file.write((char *) &st, sizeof(st));

    }
    file.close();

    //read_binary();
    //search();
    //edit2();
    read_binary();
    return 0;
}

```

پروژه:

پیاده سازی یک دفترچه تلفن در فایل متنی با قابلیت، افزودن افراد و جستجو.

```
#include <iostream>
#include <fstream>
#include <cstdlib>

using namespace std;

struct users{
    string fname, lname, phone_num;
}user;

void addContact();
void searchContact();
void help();
void self_exit();
bool check_digits(string);
bool check_numbers(string);

void self_exit()
{
    system("cls");
    cout << "\n\n\n\t\tThank You for using Contact-Saver!";
    exit(1);
}

void help(){
    cout << "Help Center";
    cout << endl << endl;
    system("pause");
    system("cls");
}

void addContact(){
    ofstream phone("number.txt", ios::app);
```

```

system("cls");
cout << "\n\n\tEnter First Name : ";
cin >> user.fname;
cout << "\n\tEnter Last Name : ";
cin >> user.lname;
cout << "\n\tEnter Phone Number : ";
cin >> user.phone_num;

if (check_digits(user.phone_num) == true) {
    if (check_numbers(user.phone_num) == true) {
        if (phone.is_open()) {
            phone << user.fname << " " << user.lname << " "
                << user.phone_num << endl;
            cout << "\n\tContact saved successfully !";
        }
        else {
            system("color 04");
            cout << "\n\tError in opening record!";
        }
    }
    else {
        system("color 04");
        cout << "\n\tOnly numbers are allowed!";
    }
}
else {
    system("color 04");
    cout << "\n\tPhone number should be of 10 digits "
        "only.";
}
cout << endl;
phone.close();
system("pause");
system("color 0A");
}

void searchContact(){
    bool found = false;
    ifstream myfile("number.txt");
    string keyword;
    cout << "\n\tEnter Name to search : ";
    cin >> keyword;
    while (myfile >> user.fname >> user.lname >> user.phone_num) {
        if (keyword == user.fname || keyword == user.lname) {
            system("cls");

```

```

        cout << "\n\n\n\t\tCONTACT DETAILS";
        cout << "\n\nFirst Name : " << user.fname;
        cout << "\nLast Name : " << user.lname;
        cout << "\nPhone Number : " << user.phone_num;
        found = true;
        break;
    }
}
if (found == false)
    cout << "\nNo such contact is found!";

cout << endl << endl;
}

bool check_digits(string x){
    if (x.length() == 10)
        return true;
    else
        return false;
}

bool check_numbers(string x){
    bool check = true;

    for (int i = 0; i < x.length(); i++)
        if (!(int(x[i]) >= 48 && int(x[i]) <= 57)) {
            check = false;
            break;
        }

    if (check == true)
        return true;

    return false;
}

int main(){
    int choice;
    system("cls");
    system("color 0A");
    while (1) {
        cout << "\n\t\t\tCONTACT SAVER";
        cout << "\n\n\t1. Add Contact\n\t2. Search "
                "Contact\n\t3. Help\n\t4. Exit\n\t> ";
        cin >> choice;
    }
}

```

```
// Switch case
switch (choice) {
case 1:
    addContact();
    break;

case 2:
    searchContact();
    break;

case 3:
    help();
    break;

case 4:
    self_exit();
    break;

default:
    cout << "\nInvalid Input!";
}
}
return 0;
}
```

پیاده سازی یک دفترچه تلفن در فایل باینری با قابلیت، افزودن، جستجو، ویرایش و حذف