

# C++: Methods of code shortening in competitive programming

Difficulty Level : Medium • Last Updated : 17 Jun, 2021

Shortcode is ideal in competitive programming because programs should be written as fast as possible. Because of this, competitive programmers often define shorter names for datatypes and other parts of the code.

We here discuss the method of code shortening in C++ specifically.

## Type names

Using the command *typedef* it is possible to give a shorter name to a datatype.

For example, the name long long is long, so we can define a shorter name ll:

```
typedef long long ll;
```

After this, the code

## CPP

```
long long a = 123456789;  
long long b = 987654321;  
cout << a * b << "\n";
```

can be shorted as follows:

## CPP



```
ll a = 123456789;
```

The command typedef can also be used with more complex types. For example, the following code gives the name vi for a vector of integers and the name pi for a pair that contains two integers,

## CPP

```
typedef vector<int> vi;  
typedef pair<int, int> pi;
```

### Macros

Another way to shorten code is to define **macros**. A macro means that certain strings in the code will be changed before the compilation. In C++, macros are defined using the #define keyword.

For example, we can define the following macros:

```
#define F first  
#define S second  
#define PB push_back  
#define MP make_pair
```

After this, the code

```
v.push_back(make_pair(y1, x1));  
v.push_back(make_pair(y2, x2));  
int d = v[i].first+v[i].second;
```

can be shorted as follows

```
v.PB(MP(y1, x1));  
v.PB(MP(y2, x2));  
int d = v[i].F+v[i].S;
```

A macro can also have parameters which makes it possible to shorten loops and others structures. For example, we can define the following macro:

```
#define REP(i, a, b) for (int i=a; i<=b; i++)
```

After this, the code

```
for (int i=1; i<=n; i++){  
    search(i);
```



```
search(i);  
}
```

## A version of template given below

This can be use in competitive programming for faster coding.

## CPP

```
#include <bits/stdc++.h> // Include every standard library  
using namespace std;  
  
typedef long long LL;  
typedef pair<int, int> pii;  
typedef pair<LL, LL> pll;  
typedef pair<string, string> pss;  
typedef vector<int> vi;  
typedef vector<vi> vvi;  
typedef vector<pii> vii;  
typedef vector<LL> vl;  
typedef vector<vl> vvl;  
  
double EPS = 1e-9;  
int INF = 1000000005;  
long long INFF = 10000000000000000005LL;  
double PI = acos(-1);  
int dirx[8] = { -1, 0, 0, 1, -1, -1, 1, 1 };  
int diry[8] = { 0, 1, -1, 0, -1, 1, -1, 1 };  
  
#ifdef TESTING  
#define DEBUG fprintf(stderr, "====TESTING====\n")  
#define VALUE(x) cerr << "The value of " << #x << " is " << x << endl  
#define debug(...) fprintf(stderr, __VA_ARGS__)  
#else  
#define DEBUG  
#define VALUE(x)  
#define debug(...)  
#endif  
  
#define FOR(a, b, c) for (int(a) = (b); (a) < (c); ++(a))  
#define FORN(a, b, c) for (int(a) = (b); (a) <= (c); ++(a))  
#define FORD(a, b, c) for (int(a) = (b); (a) >= (c); --(a))  
#define FORSQ(a, b, c) for (int(a) = (b); (a) * (a) <= (c); ++(a))  
#define FORC(a, b, c) for (char(a) = (b); (a) <= (c); ++(a))  
#define FOREACH(a, b) for (auto&(a) : (b))  
#define REP(i, n) FOR(i, 0, n)  
#define REPN(i, n) FORN(i, 1, n)
```

```

#define RESET(a, b) memset(a, b, sizeof(a))
#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define ALL(v) v.begin(), v.end()
#define ALLA(arr, sz) arr, arr + sz
#define SIZE(v) (int)v.size()
#define SORT(v) sort(ALL(v))
#define REVERSE(v) reverse(ALL(v))
#define SORTA(arr, sz) sort(ALLA(arr, sz))
#define REVERSEA(arr, sz) reverse(ALLA(arr, sz))
#define PERMUTE next_permutation
#define TC(t) while (t--)

```

```

inline string IntToString(LL a)
{
    char x[100];
    sprintf(x, "%lld", a);
    string s = x;
    return s;
}

```

```

inline LL StringToInt(string a)
{
    char x[100];
    LL res;
    strcpy(x, a.c_str());
    sscanf(x, "%lld", &res);
    return res;
}

```

```

inline string GetString(void)
{
    char x[1000005];
    scanf("%s", x);
    string s = x;
    return s;
}

```

```

inline string uppercase(string s)
{
    int n = SIZE(s);
    REP(i, n)
        if (s[i] >= 'a' && s[i] <= 'z')
            s[i] = s[i] - 'a' + 'A';
    return s;
}

```



```

REP(i, n)
if (s[i] >= 'A' && s[i] <= 'Z')
    s[i] = s[i] - 'A' + 'a';
return s;
}

```



## Related Articles

```

// end of Sektor_jr template v2.0.3 (BETA)

int main()
{
    freopen("A.in", "r", stdin);
    freopen("output.txt", "w", stdout);

    int a, b;
    fin >> a >> b;
    fout << a + b << endl;
    return 0;
}

```

Attention reader! Don't stop learning now. Get hold of all the important Comcompetitivepetitve Programming concepts with the [Competitive Programming Live](#) course.

Like 52



Previous

Next

## Article Contributed By :



GeeksforGeeks

## Vote for difficulty

Current difficulty : [Medium](#)

Easy

Normal

Medium

Hard

Expert

Article Tags : [cpp-macros](#), [C Language](#), [C++](#), [Competitive Programming](#)

Practice Tags : [CPP](#)

Improve Article

Report Issue

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

Load Comments



GeeksforGeeks

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

About Us  
Careers  
Privacy Policy  
Contact Us  
Copyright Policy

Algorithms  
Data Structures  
Languages  
CS Subjects  
Video Tutorials

## Practice

Courses  
Company-wise  
Topic-wise  
How to begin?

## Contribute

Write an Article  
Write Interview Experience  
Internships  
Videos

@geeksforgeeks , Some rights reserved

