DIRECTIVE PREPROCESOR #DEFINE

CONSTANTE SIMBOLICE

```
// definire constanta
#define DIM_VECTOR 20
// definire tip de data
#define ULL unsigned long long
// definire mesaj
#define MESAJ "Calcul suma"
// definire cod pe mai multe linii
#define SEPARATOR cout \
<< "---------------" \
<< endl;
```

CONSTANTE SIMBOLICE - EXEMPLU

```
#include <bits/stdc++.h>
using namespace std;
#define DIM_VECTOR 20
#define ull unsigned long long
#define MESAJ "Calcul suma"
#define SEPARATOR cout << "\n----------------\n";
int main()
{ ull a=5, b=7, v[DIM_VECTOR];
  cout << MESAJ; SEPARATOR
  cout << a; SEPARATOR
  cout << b;
  return 0;
}
```

MACRODEFINIȚII

- Forma: #define macro(parametri) corp
- Exemplu:
- #define s(x, y) y*(y+1)/2 – x*(x-1)/2
- Ce face: suma numerelor din intervalul [x, y]

MACRODEFINIȚII

```
#include <iostream>
using namespace std;
#define s(x,y) y*(y+1)/2-x*(x-1)/2
int a, b, c;
int main()
{
   cin >> a >> b >> c;
   cout << s(a, b) << " " << s(b, c) << " " << s(a, c);
   return 0;
}
```

MACRODEFINIȚII

```
#include <bits/stdc++.h>
using namespace std;
#define CICLU(NR_ITERATII,CORP) \
for (int i = 0; i < NR_ITERATII; i++) { CORP; }
int main()
{ int s;
  CICLU(10, s+=i;);cout << s;
  CICLU(10,cout << i<< " "; s++;); cout << s;
  return 0;
}
```

MACRODEFINITII

```
#include <bits/stdc++.h>
using namespace std;
#define endl '\n'
#define watch(x) cout << (#x) << " = " << (x) << endl
int main()
{ int a=5, b=7;
  watch(a);
  return 0;
}
```

MACRODEFINITII

```
#include <bits/stdc++.h>
using namespace std;
#define tot(x) (x).begin(), (x).end()
#define citeste(x,n)\
 for(int a,i=0; i<n; i++){cin >> a; (x).push_back(a);}
#define afis(x) for(auto i:(x))cout << i << " ";
int main()
{ vector<int>v;
  citeste(v,5);
  sort(tot(v));
  afis(v);
  return 0;
}
```