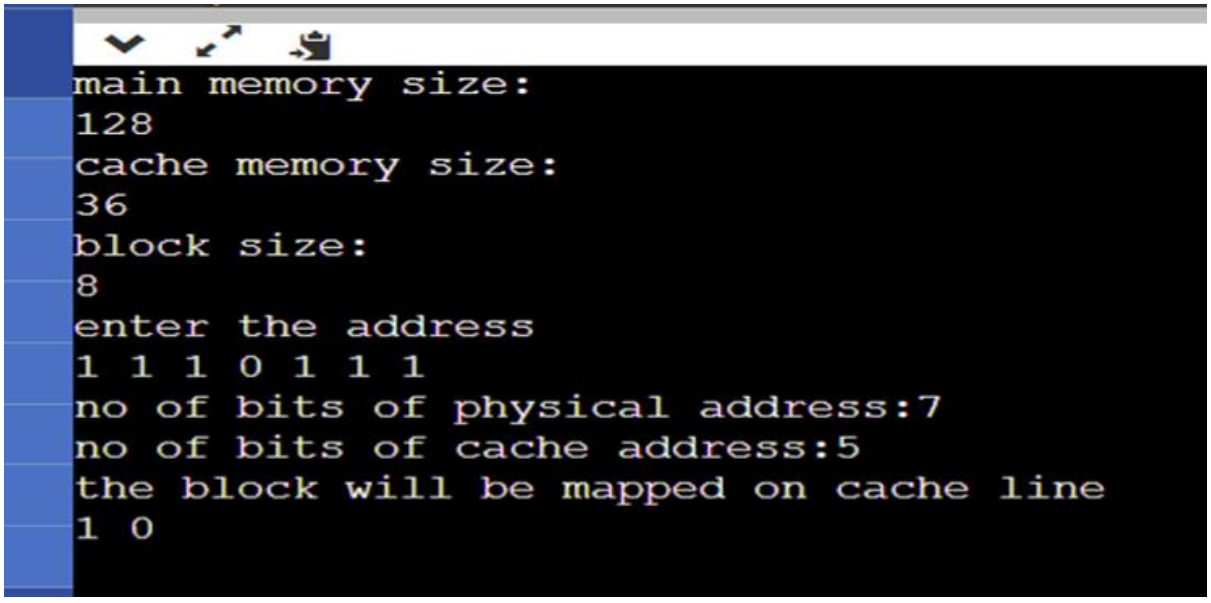
Memory Mapping

```
 #include <iostream>
#include <bits/stdc++.h>
using namespace std;
int power_2(int a)
{
   return log(a)/log(2);
}
int main()
{
   //input
   int msize,bsize,csize;
   cout<<"main memory size: "<<endl;
   cin>>msize;
 cout<<"cache memory size: "<<endl;
 cin>>csize;
  cout<<"block size: "<<endl;
  cin>>bsize;
  //initialization
  int physicaladd=power_2(msize);
  int cacheadd = power_2(csize);

  int num_blocks=msize/bsize;
  int lines= csize/bsize;
  int block_bits=power_2(bsize);
  int linebits=power_2(lines);
  //mapping
  vector<int>address(physicaladd,0);
  cout<<"enter the address"<<endl;
  for(int i=0;i<physicaladd;i++)
```

```
  {
     int x;
     cin>>x;
     address[i]=x;
  }
  int msb = physicaladd-block_bits;
  vector<int>Msb(msb,0);
  for(int i=0;i<msb;i++)
  {
     Msb[i]=address[i];
  }
  int req = msb-linebits;
  vector<int>result;
  for(int i=req;i<msb;i++)
  {
     result.push_back(Msb[i]);
  }
  cout<<"no of bits of physical address:"<<physicaladd<<endl;
  cout<<"no of bits of cache address:"<<cacheadd<<endl;
  cout<<"the block will be mapped on cache line"<<endl;
  for(int i=0;i<req;i++)
  cout<<result[i]<<" ";

    return 0;
}
```


main memory size:
128
cache memory size:
36
block size:
8
enter the address
1 1 1 0 1 1 1
no of bits of physical address:7
no of bits of cache address:5
the block will be mapped on cache line
1 0