1. First fit memory allocation

**Input:**

#include <iostream>

using namespace std;

int main()

{

  int pno,bno,psize[10],bsize[10],flag[10],allocation[10];

  cout<<"Enter number of processes:";

  cin>>pno;

  cout<<"Enter process sizes:"<<endl;

  for(int i=0;i<pno;i++)

  {

    cout<<"Process "<<(i+1)<<": ";

    cin>>psize[i];

  }

  cout<<endl;

  cout<<"Enter number of blocks:";

  cin>>bno;

  cout<<"Enter block sizes:"<<endl;

  for(int i=0;i<bno;i++)

  {

    cout<<"Block "<<(i+1)<<": ";

    cin>>bsize[i];

    flag[i]=0;

    allocation[i]=-1;

  }

for(int i=0;i<pno;i++)

{

for(int j=0;j<bno;j++)

{

  if(flag[j]==0 && bsize[j]>=psize[i])

  {

    allocation[i]=j;

    flag[j]=1;

    break;

  }

}

  }

  cout<<"\nProcess number"<<"\t"<<"Process size"<<"\t"<<"Block number"<<"\t"<<"Block size\n";

  for(int i=0;i<pno;i++)

  {

    cout<<"\nP"<<i+1<<"\t\t"<<psize[i];

    if(allocation[i]!=-1)

    {

      cout<<"\t\tB"<<(allocation[i]+1)<<"\t\t"<<bsize[allocation[i]];

    }

    else

    {

      cout<<"\t\tUnallocated";

    }

  }

  return 0;

}

**Output:**

Enter number of processes:4

Enter process sizes:

Process 1: 200

Process 2: 50

Process 3: 150

Process 4: 100

Enter number of blocks:4

Enter block sizes:

Block 1: 500

Block 2: 300

Block 3: 90

Block 4: 150

Process number Process size Block number Block size

P1 200 B1 500

P2 50 B2 300

P3 150 B4 150

P4 100 Unallocated

1. Worst fit memory allocation

**Input:**

#include <iostream>

using namespace std;

int main()

{

  int pno,bno,temp=0,psize[10],bsize[10],flag[10],allocation[10],big=0;

  cout<<"Enter number of processes:";

  cin>>pno;

  cout<<"Enter process sizes:"<<endl;

  for(int i=0;i<pno;i++)

  {

    cout<<"Process "<<(i+1)<<": ";

    cin>>psize[i];

  }

      cout<<endl;

      cout<<"Enter number of blocks:";

      cin>>bno;

      cout<<"Enter block sizes:"<<endl;

  for(int i=0;i<bno;i++)

  {

    cout<<"Block "<<(i+1)<<": ";

    cin>>bsize[i];

    flag[i]=0;

    allocation[i]=-1;

  }

  for(int i=0;i<pno;i++)

  {

    for(int j=0;j<bno;j++)

    {

        if(flag[j] != 1)

        {

            temp = bsize[j]-psize[i];

              if (temp >= 0)

              {

                 if (big <= temp)

                 {

                    allocation[i]=j;

                    big=temp;

                 }

              }

        }

    }

    flag[allocation[i]]=1;

    big=0;

  }

  cout<<"\nProcess number"<<"\t"<<"Process size"<<"\t"<<"Block  number"<<"\t"<<"Block size\n";

  for(int i=0;i<bno;i++)

  {

        cout<<"\nP"<<i+1<<"\t\t"<<psize[i];

    if(allocation[i]  == -1)

     {

      cout<<"\t\tUnallocated";

     }

    else

     {

       cout<<"\t\tB"<<(allocation[i]+1)<<"\t\t"<<bsize[allocation[i]];

     }

  }

  return 0;

}

**Output:**

Enter number of processes:4

Enter process sizes:

Process 1: 200

Process 2: 50

Process 3: 150

Process 4: 100

Enter number of blocks:4

Enter block sizes:

Block 1: 500

Block 2: 300

Block 3: 90

Block 4: 140

Process number Process size Block number Block size

P1 200 B1 500

P2 50 B2 300

P3 150 Unallocated

P4 100 B4 140

1. Best fit memory allocation

**Input:**

#include <iostream>

using namespace std;

int main()

{

  int pno,bno,temp,psize[10],bsize[10],flag[10],allocation[10],small=999;

  cout<<"Enter number of processes:";

  cin>>pno;

  cout<<"Enter process sizes:"<<endl;

  for(int i=0;i<pno;i++)

  {

    cout<<"Process "<<(i+1)<<": ";

    cin>>psize[i];

  }

      cout<<endl;

      cout<<"Enter number of blocks:";

      cin>>bno;

      cout<<"Enter block sizes:"<<endl;

  for(int i=0;i<bno;i++)

  {

    cout<<"Block "<<(i+1)<<": ";

    cin>>bsize[i];

    flag[i]=0;

    allocation[i]=-1;

  }

  for(int i=0;i<pno;i++)

  {

    for(int j=0;j<bno;j++)

    {

        if(flag[j] != 1)

        {

            temp = bsize[j]-psize[i];

              if (temp >= 0)

              {

                 if (small >= temp)

                 {

                    allocation[i]=j;

                    small=temp;

                 }

              }

        }

    }

    flag[allocation[i]]=1;

    small=999;

  }

  cout<<"\nProcess number"<<"\t"<<"Process size"<<"\t"<<"Block number"<<"\t"<<"Block size\n";

  for(int i=0;i<bno;i++)

  {

        cout<<"\nP"<<i+1<<"\t\t"<<psize[i];

    if(flag[i]==1)

     {

       cout<<"\t\tB"<<(allocation[i]+1)<<"\t\t"<<bsize[allocation[i]];

     }

    else

     {

      cout<<"\t\tUnallocated";

     }

  }

  return 0;

}

**Output:**

Enter number of processes:4

Enter process sizes:

Process 1: 200

Process 2: 50

Process 3: 150

Process 4: 100

Enter number of blocks:4

Enter block sizes:

Block 1: 500

Block 2: 300

Block 3: 90

Block 4: 150

Process number Process size Block number Block size

P1 200 B2 300

P2 50 B3 90

P3 150 B4 150

P4 100 B1 500