**Problem 1893 内存管理**

**Accept: 63 Submit: 214  
Time Limit: 1000 mSec Memory Limit : 32768 KB**

**http://acm.fzu.edu.cn/image/prodesc.gifProblem Description**

在为进程动态分配内存时,操作系统必须对其进行管理。一种管理内存的方式是，维护一张记录内存使用情况的表来模拟内存的存储情况，其任一表项可能是以下两种情况之一：一个已分配给进程的内存区或者一个空闲的内存区。表中每一个表项含有以下内容：空闲区或已使用内存区的指示标志、内存大小、指向下一表项的指针。

一个表项一般有一个前驱表项和一个后继表项（除非这个表项是内存的最底端或最顶端），已使用区表项的前驱表项和后继表项可能是已使用区表项也可能是空闲区表项，而空闲区表项的前驱表项和后继表项只可能是已使用区表项，因为如果两个空闲区表项如果互为相邻的表项，我们就可以将它们合并为一个空闲区表项，内存大小为两个空闲区表项之和。

当系统创建一个新的进程时，有很多种方法可以用来为新创建的进程分配内存，在这里我们假设已知新进程要占用的内存大小，并只可能给进程分配连续的内存空间。在本题中，我们采用最佳适配算法来分配内存。最佳适配算法搜索整个表，找出够用的最小的空闲区（如有多个，取表中最前的），并将新进程分配在空闲区的顶部，如有剩余内存仍作为空闲区表项，并且为新已使用区表项的下一表项。

当系统结束一个进程时，在整个表中找到相应的已使用区表项并将其变为空闲区表项。

你的任务是编写一个程序，来实现这样一个内存管理的过程。在本题中,初始时内存表只有一个内存空闲区表项，且内存大小为100个单位大小。

**http://acm.fzu.edu.cn/image/prodesc.gifInput**

输入数据第一行为一整数T，表示有T组输入数据。

每组数据每行是一个系统命令，包括“Create”,“Delete”,“Print”，“End”，每条命令后面的参数如下：

Create：两个整数i和size，i表示新进程的进程号，i>0，size表示新进程需占用的内存大小，size>0，输入保证新进程的进程号与已存在的进程号不同。

Delete：一个整数i，i>0，表示要结束的进程号。

Print：无参数。

End：无参数，不做任何处理，表示系统终止命令。

注意：输入保证出现的表项个数不超过100个。

**http://acm.fzu.edu.cn/image/prodesc.gifOutput**

对于Create命令，如果成功创建新进程，输出一行“Create process i of size s successfully!”，其中i为新进程的进程号，s为新进程占用的内存大小；如果无足够大小的内存空闲区，输出一行“No enough memory!”。

对于Delete命令，如果成功结束进程，输出一行“Delete process i of size s successfully!”，其中i为结束进程的进程号，s为该进程占用的内存大小；如果内存中无该进程，输出一行“No such process!”。

对于Print命令，将内存表从表头开始按顺序输出，每行输出一个表项，进程表项格式为“P 进程号 内存大小”，空闲区表项格式为“H 内存大小”。

对于End命令，不做任何处理，终止程序。

**http://acm.fzu.edu.cn/image/prodesc.gifSample Input**

2

Create 1 30

Create 2 20

Create 3 30

Print

Create 4 100

Delete 4

Delete 2

Print

Delete 3

Print

End

Create 1 100

Print

End

**http://acm.fzu.edu.cn/image/prodesc.gifSample Output**

Create process 1 of size 30 successfully!

Create process 2 of size 20 successfully!

Create process 3 of size 30 successfully!

P 1 30

P 2 20

P 3 30

H 20

No enough memory!

No such process!

Delete process 2 of size 20 successfully!

P 1 30

H 20

P 3 30

H 20

Delete process 3 of size 30 successfully!

P 1 30

H 70

Create process 1 of size 100 successfully!

P 1 100

**http://acm.fzu.edu.cn/image/prodesc.gifSource**

福州大学第七届程序设计竞赛

#include<iostream>

#include<vector>

#include<string>

using namespace std;

class node

{

public:

node\* next;

node\* pre;

int size;

int id;

};

int main()

{

int t,i;

char s[10];

int id,size;

scanf("%d",&t);

while(t--)

{

node\* first=new node,\*min;

first->next=0;

first->pre=0;

first->size=100;

first->id=0;

vector<node\*> b;

node\* p=first;

bool flag;

while(scanf("%s",s) && s[0]!='E')

{

if(s[0]=='C')

{

scanf("%d%d",&id,&size);

flag=false;

p=first;

while(p)

{

if(!p->id && p->size>=size)

{

b.push\_back(p) ;

flag=true;

}

p=p->next;

}

if(flag)

{

min=b[0];

for(i=1;i<b.size();i++)

if(min->size>b[i]->size)

min=b[i];

node\* q=new node;

q->id=id;

q->size=size;

q->next=min;

min->size-=size;

if(min==first)

{

first=q;

min->pre=q;

}

else

{

q->pre=min->pre;

min->pre->next=q;

min->pre=q;

}

if(min->size==0)

{

if(min->next)

{

min->next->pre=q;

q->next=min->next;

}

else

min->pre->next=0;

delete min;

}

printf("Create process %d of size %d successfully!\n",id,size);

}

else

printf("No enough memory!\n");

}

if(s[0]=='D')

{

flag=false;

scanf("%d",&id);

p=first;

while(p)

{

if(p->id==id)

{

flag=true;

size=p->size;

if(p->next && p!=first && !p->next->id && !p->pre->id)

{

p->pre->size+=p->size+p->next->size;

p->pre->next=p->next->next;

if(p->next->next)

p->next->next->pre=p->pre;

delete p;

}

else

if(p->next && !p->next->id)

{

p->next->size+=p->size;

if(p==first)

{first=p->next;}

else

{

p->pre->next=p->next;

p->next->pre=p->pre;

}

delete p;

}

else

if(p->pre && !p->pre->id)

{

p->pre->size+=p->size;

p->pre->next=p->next;

if(p->next)

p->next->pre=p->pre;

delete p;

}

else

p->id=0;

break;

}

p=p->next;

}

if(flag)

printf("Delete process %d of size %d successfully!\n",id,size);

else

printf("No such process!\n");

}

if(s[0]=='P')

{

p=first;

while(p)

{

if(p->id)

printf("P %d %d\n",p->id,p->size);

else

printf("H %d\n",p->size);

p=p->next;

}

}

}

}

return 0;

}