1. #include<iostream>
2. **using** **namespace** std;
3. #include<vector>
4. #include<string>
5. #include<algorithm>



10. **struct** Call
11. {
12. string name;
13. **int** month;
14. **int** date;
15. **int** hour;
16. **int** minute;
17. **int** total;
18. string status;
19. };
21. **int** charge[24];
22. vector<Call> all\_calls;
23. vector<Call> format\_calls;
25. **bool** compare(Call a,Call b)
26. {
27. **if**(a.name < b.name)
28. **return** 1;
29. **else** **if**(a.name == b.name && a.total < b.total)
30. **return** 1;
31. **else**
32. **return** 0;
33. }

//calculate money from the begin of month to the time given

1. **int** chargeByTime(**int** time)
2. {
3. **int** hours = time/60;
4. **int** minutes = time%60;
5. **int** money = 0;
6. **int** i;
7. **for**(i = 0;i<hours;i++)
8. money += charge[i%24]\*60;
9. money += charge[i%24]\*minutes;
10. **return** money;
11. }
13. **double** calCost(Call s,Call t)
14. {
15. **return** (**double**)(chargeByTime(t.total)-chargeByTime(s.total))/100;
16. }
18. **int** calLast(Call s,Call t)
19. {
20. **return** (t.date-s.date)\*24\*60+(t.hour-s.hour)\*60+(t.minute-s.minute);
21. }


25. **int** main()
26. {
27. **for**(**int** i = 0;i<24;i++)
28. cin>>charge[i];
29. **int** N;
30. cin>>N;
31. **while(N--)**
32. **{**
33. **Call c;**
34. **cin>>c.name;**
35. **cin>>c.month;**
36. **getchar();**
37. **cin>>c.date;**
38. **getchar();**
39. **cin>>c.hour;**
40. **getchar();**
41. **cin>>c.minute;**
42. **c.total = c.minute + 60\*c.hour + 24\*60\*c.date;**
43. **cin>>c.status;**
44. **all\_calls.push\_back(c);**
45. **}**
46. sort(all\_calls.begin(),all\_calls.end(),compare);
48. //filter delete those bad record
49. **bool** haveonline = **false**;
50. string curname;
51. **for**(**int** i=0;i<all\_calls.size();i++)
52. {
53. **if**(haveonline == **false** && all\_calls[i].status =="on-line" )
54. {
55. format\_calls.push\_back(all\_calls[i]);
56. haveonline = **true**;
57. curname = all\_calls[i].name;
58. }
59. **else** **if**(haveonline == **true** && all\_calls[i].status =="on-line")
60. {
61. format\_calls.pop\_back();
62. format\_calls.push\_back(all\_calls[i]);
63. haveonline = **true**;
64. curname = all\_calls[i].name;
65. }
66. **else** **if**(haveonline == **true** && all\_calls[i].status =="off-line"&&all\_calls[i].name ==curname)
67. {
68. format\_calls.push\_back(all\_calls[i]);
69. haveonline = **false**;
70. }
71. }
72. //the last must be offline
73. **if**((\*(format\_calls.end()-1)).status == "on-line")
74. format\_calls.pop\_back();
76. //output
77. **double** totalcost = 0;
78. curname = "";
79. **for**(**int** i=0;i<format\_calls.size();i+=2)
80. {
82. **if**(format\_calls[i].name != curname)
83. {
84. **if**(curname!="")
85. {
86. printf("Total amount: $%.2f\n",totalcost);
87. totalcost = 0;
88. printf("%s %02d\n",format\_calls[i].name.c\_str(),format\_calls[i].month);
89. }
90. **else**
91. {
92. printf("%s %02d\n",format\_calls[i].name.c\_str(),format\_calls[i].month);
93. }
94. curname = format\_calls[i].name;
95. }
96. printf("%02d:%02d:%02d",format\_calls[i].date,format\_calls[i].hour,format\_calls[i].minute);
97. printf(" ");
98. printf("%02d:%02d:%02d",format\_calls[i+1].date,format\_calls[i+1].hour,format\_calls[i+1].minute);
99. printf(" ");
100. printf("%d",calLast(format\_calls[i],format\_calls[i+1]));
101. printf(" ");
102. printf("$%.2f\n",calCost(format\_calls[i],format\_calls[i+1]));
103. totalcost+=calCost(format\_calls[i],format\_calls[i+1]);
104. }
105. printf("Total amount: $%.2f\n",totalcost);

108. }

#include**<cmath>**

#include**<cstdio>**

#include**<iostream>**

#include**<string>**

#include**<algorithm>**

#include**<queue>**

#include**<vector>**

#include**<sstream>**

**using** **namespace** std;

**struct** call {

**string** name;

**int** month;

**int** day;

**int** hour;

**int** minute;

**int** total = 0;

**string** status;

};

**int** charge[24];

vector<call>all\_calls;

vector<call>selected\_calls;

**int** comp(call a, call b) {

**if** (a.name < b.name)

{

**return** 1;

}

**else** **if** (a.name == b.name&&a.total < b.total)

{

**return** 1;

}

**else**

**return** 0;

}

**double** chargeFound(**int** time)

{

**int** hour, mins, money = 0;

hour = time / 60;

mins = time % 60;

**for** (**int** i = 0; i < hour; i++)

money += charge[i % 24] \* 60;

money += mins\*charge[hour % 24];

**return** money;

}

**double** chargeamount(call s, call e)

{

**return** (chargeFound(e.total) - chargeFound(s.total))/100;

}

**int** main()

{

**int** num;

//record;

**for** (**int** i = 0; i < 24; i++)

scanf("%d", &charge[i]);

scanf("%d", &num);

**for** (**int** i = 0; i < num; i++)

{

call temp;

cin >> temp.name;

getchar();

cin >> temp.month;

getchar();

cin >> temp.day;

getchar();

cin >> temp.hour;

getchar();

cin >> temp.minute;

getchar();

cin >> temp.status;

temp.total = temp.minute + temp.hour \* 60 + temp.day \* 24 \* 60;

all\_calls.push\_back(temp);

}

//sort

sort(all\_calls.begin(), all\_calls.end(), comp);

//select

**bool** haveonline = false;

**string** curname;

**for** (**int** i = 0; i < all\_calls.size(); i++)

{

**if** (haveonline == false && all\_calls[i].status == "on-line")

{

selected\_calls.push\_back(all\_calls[i]);

curname = all\_calls[i].name;

haveonline = true;

}

**else** **if** (haveonline == true && all\_calls[i].status == "on-line")

{

selected\_calls.pop\_back();

selected\_calls.push\_back(all\_calls[i]);

haveonline = true;

curname = all\_calls[i].name;

}

**else** **if** (haveonline == true && all\_calls[i].status == "off-line"&&curname == all\_calls[i].name)

{

selected\_calls.push\_back(all\_calls[i]);

haveonline = false;

}

}

**if** ((\*(selected\_calls.end() - 1)).status == "on-line")

selected\_calls.pop\_back();

//output

curname = "";

**double** totalamount = 0;

**for** (**int** i = 0; i < selected\_calls.size(); i += 2)

{

**if** (selected\_calls[i].name != curname)

{

**if** (curname == "") {

printf("%s %02d\n", selected\_calls[i].name.c\_str(), selected\_calls[i].month);

}

**else** {

printf("Total amount: $%.2f\n", totalamount);

totalamount = 0;

printf("%s %02d\n", selected\_calls[i].name.c\_str(), selected\_calls[i].month);

}

curname = selected\_calls[i].name;

}

printf("%02d:%02d:%02d ", selected\_calls[i].day, selected\_calls[i].hour, selected\_calls[i].minute);

printf("%02d:%02d:%02d ", selected\_calls[i+1].day, selected\_calls[i+1].hour, selected\_calls[i+1].minute);

printf("%d ", selected\_calls[i + 1].total - selected\_calls[i].total);

printf("$%.2f\n", chargeamount(selected\_calls[i], selected\_calls[i + 1]));

totalamount += chargeamount(selected\_calls[i], selected\_calls[i + 1]);

}

printf("Total amount: $%.2f", totalamount);

**return** 0;

}

