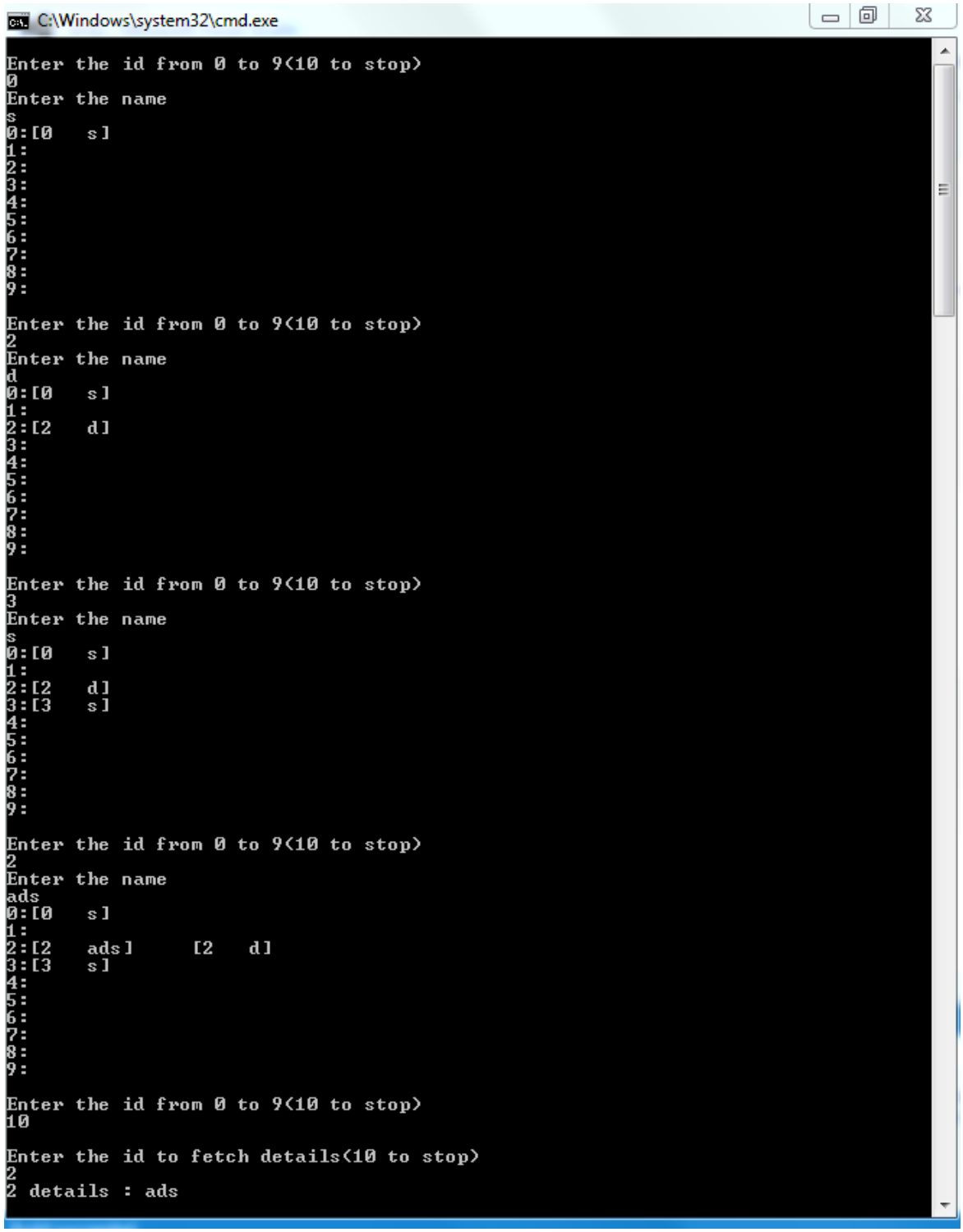
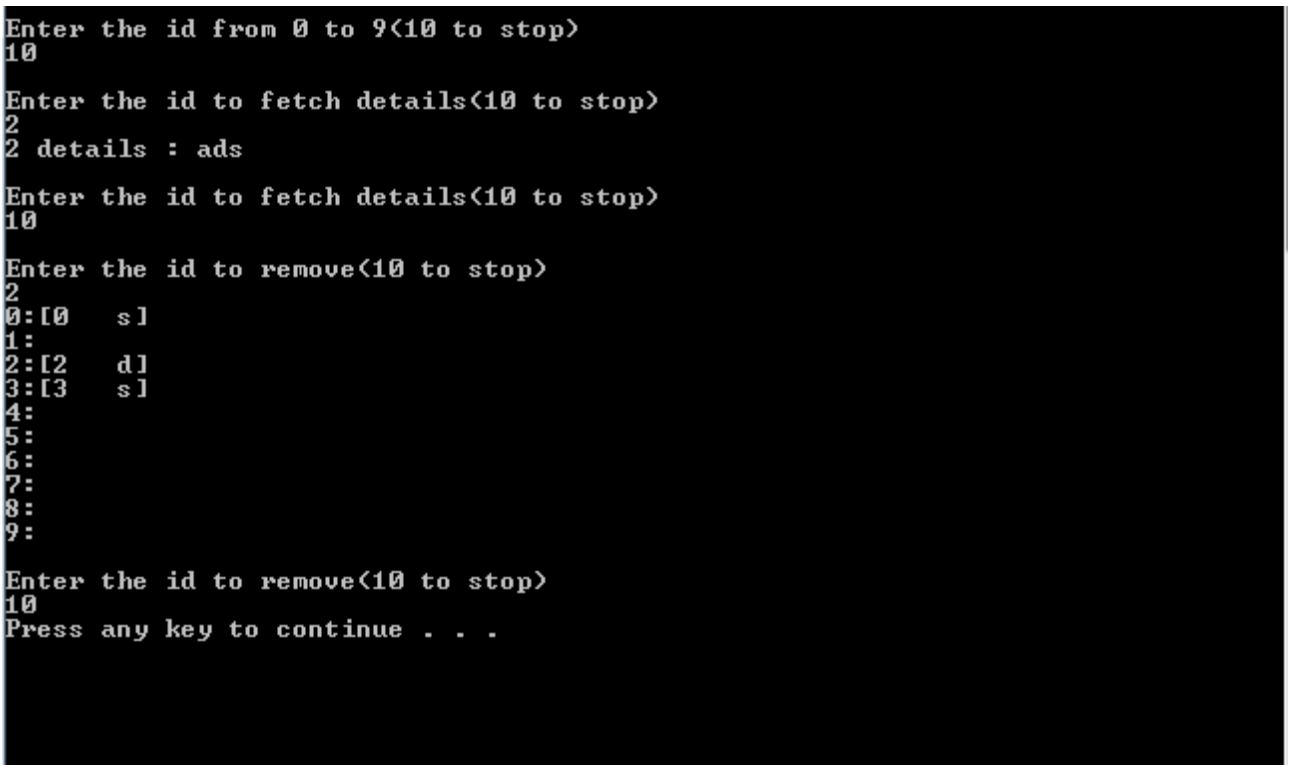
```
#include <iostream>
#include <string>
using namespace std;

//string library for entering name
class Hashtable
{
	struct Entry
	{
		int key;
		string name;
		Entry *next;
		Entry(int key, const string name): key(key), name(name), next(nullptr)
		{}
	};
	Entry **table;
	int size;

	//creating hash
	int Hash(int key)
	{
		int offset= key%10;
		return offset;
	}
public:

	//creating hashtable in heap with initializing each column by null
	Hashtable(int s=10)
	{
		size=s;
		table = new Entry*[size];
		for(int offset =0; offset<size; offset++)
		{
			table[offset]=nullptr;
		}
	}

	//Entering id and name into each column in hashtable
	void put(int key, const string &name)
	{
		int offset=Hash(key);
		Entry *newEntry = new Entry(key,name);
		newEntry->next=table[offset];
		table[offset]=newEntry;
	}

	//displaying the hashtable
	void printdebug()
	{
		for(int i=0;i<size;i++)
		{
			cout<<i<<":";
			for(Entry *current=table[i];current !=nullptr;current=current->next)
			{
				cout<<"["<<current->key<<"   "<<current->name<<"]"<<"\t";

			}
			cout<<endl;
		}

	}

	//fetching details from given id
	bool get(int key)
	{
```

```
		int offset=Hash(key);

		for( Entry *current=table[offset]; current; current=current->next)
		{
			if(key == current->key)
			{
				cout<<current->key<<" details : "<<current->name<<endl;
				return true;
			}
			cout<<"not found"<<endl;
			return false;
		}
		cout<<"not found"<<endl;
		return false;
	}

	//removing the data from given id
	void remove(int key)
	{
		int offset = Hash(key);
		//assigning both current and back to same element in hashtable
		Entry* back=table[offset];
		for(Entry*current=table[offset]; current; current->next)
		{
			if(key == current->key)
			{
				//if the given element is first one in the hashmap
				if(current == table[offset])
				{
				table[offset]=current->next;
				delete current;
				return;
				}
				else
				{
				back->next = current->next;
				delete current;
				return;
				}
			}
			back=current;
		}
	}
};
int main()
{
	int id;
	string name;
	Hashtable H;
	while(cout<<endl<<"Enter the id from 0 to 9(10 to stop)"<<endl,
		cin>>id,
		id!=10)
	{
		cout<<"Enter the name"<<endl;
		cin>>name;
		H.put(id,name);
		H.printdebug();
	}

	while(cout<<endl<<"Enter the id to fetch details(10 to stop)"<<endl,
		cin>>id,
		id!=10)
	{
		H.get(id);
	}
```

```
        while(cout<<endl<<"Enter the id to remove(10 to stop)"<<endl,
              cin>>id,
              id!=10)
        {
        H.remove(id);
        H.printdebug();
        }

        return 0;
}
```

OUTPUT:




Enter the id from 0 to 9(10 to stop)
10
Enter the id to fetch details(10 to stop)
2
2 details : ads
Enter the id to fetch details(10 to stop)
10
Enter the id to remove(10 to stop)
2
0:[0 s]
1:
2:[2 d]
3:[3 s]
4:
5:
6:
7:
8:
9:
Enter the id to remove(10 to stop)
10
Press any key to continue . . .