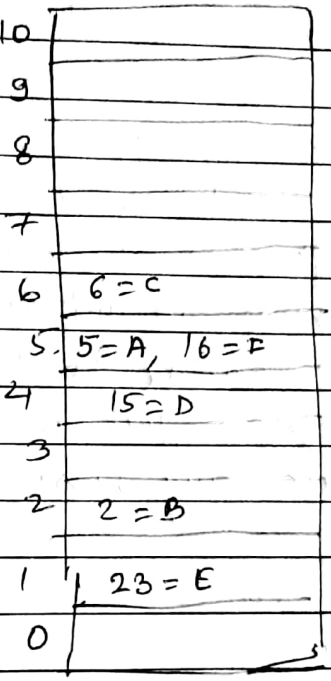
```
public int hashcode()
{
    return i;
}
public String toString()
{
    return i + " ";
}
}
```

if we are returning i%9 then o/p will be changed.

o/p ⇒ { 16=F, 15=D, 6=C, 23=E, 5=A, 2=B }

Q) D.I.C = 11 (for H/M

o/p ⇒ {6=C, 16=F, 5=A, 15=D, 2=B, 23=E }

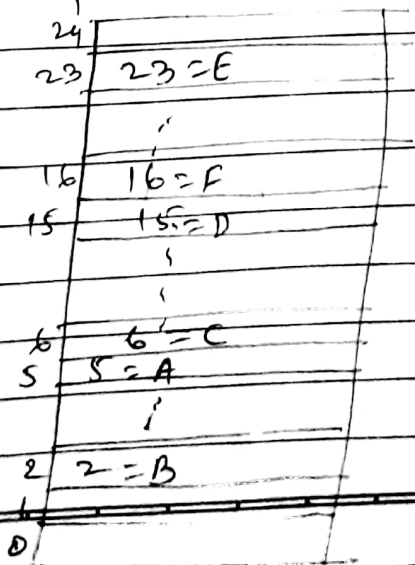


From Top to buttom and From Right to left

⇒ In the above prog, if we change the HashTable implementation as

HashTable h = new HashMap(25);

then o/p is ⇒ {23=E, 16=F,



=> Properties :->

→ In our prog, if any thing which changes frequently (like username, password, mailid, mob no etc) are not recommandate to hard code in java prog. because if there is any change to reflect that change recompilation, rebuild, and redeploy application are required. even sometimes server restart also required. which creates a big business impact to the client.

We can overcome this problem by using properties file. Such type of variable things we have to configure in the properties file. From that properties file, we have to read into java programme and we can use those properties.
The main advantage of this approache is if there is a change in properties file, to reflect that change., just re-deployment in enough which won't create any business impact to the client.

→ We can use java properties objects to hold properties which are coming from properties file.

→ In normal map (like HashMap, HashTable, TreeMap) key and value can be any type. but, in the case of properties key and value should be of String type.

Constructor :-

① properties p = new properties();

Methods :-

① String setProperty(String pname, String pvalue)
↳ to set a new property.

→ If the specified property already available, then old value