Started on	Saturday, 3 December 2016, 3:40 PM
State	Finished
Completed on	Saturday, 3 December 2016, 3:41 PM
Time taken	1 min
Marks	4.00/5.00
Grade	**80.00** out of 100.00

Question 1

Correct Mark 1.00 out of 1.00

What would be the output of the following code if quadratic probing policy is used?

```
table = [0]*10
def myhash(x):return x%10

def insert(table, value):
    table[myhash(value)]=value #This line is changed with quadratic probing policy

insert(table, 3)
insert(table, 4)
insert(table, 13)
insert(table, 33)
print table
```

Select one:

a.

```
[0, 0, 0, 3, 4, 13, 33, 0, 0, 0]
```

b.

```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

c.

```
[0, 0, 0, 3, 4, 0, 0, 13, 0, 33]
```

d.

```
[0, 0, 33, 3, 4, 0, 0, 13, 0, 0]
```

✓

Your answer is correct.

The correct answer is:

```
[0, 0, 33, 3, 4, 0, 0, 13, 0, 0]
```

Question 2

Correct Mark 1.00 out of 1.00

What would be the output of the following code if linear probing policy is used?

```
table = [0]*10
def myhash(x):return x%10

def insert(table, value):
    table[myhash(value)]=value #This line is changed with linear probing policy

insert(table, 3)
```

insert(table, 4)

insert(table, 13)

print table

Select one:

a.

```
[0, 0, 0, 13, 4, 0, 0, 0, 0, 0]
```

b.

```
[0, 0, 13, 3, 4, 0, 0, 0, 0, 0]
```

c.

```
[0, 0, 0, 3, 4, 13, 0, 0, 0, 0]
```

d.

```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

Your answer is correct.

The correct answer is:

```
[0, 0, 0, 3, 4, 13, 0, 0, 0, 0]
```

Question 3

Incorrect Mark 0.00 out of 1.00

Which one of the following is not a collision resolution policy?

Select one:

- a. Chaining
- b. Double Hashing
- c. Open Addressing ✗
- d. Linear Hashing

Your answer is incorrect.

The correct answer is: Linear Hashing

Question 4

Correct Mark 1.00 out of 1.00

Hash tables associate a key to a value, that is why they are sometimes called _____.

Select one:

- a. arrays
- b. chains
- c. None of them
- d. associative arrays ✓

Your answer is correct.

The correct answer is: associative arrays

Question 5

Correct Mark 1.00 out of 1.00

What would be the output of the following code if quadratic probing policy is used?

```
table = [0]*10
def myhash(x):return x%10

def insert(table, value):
    table[myhash(value)]=value #This line is changed with quadratic probing policy

insert(table, 3)
insert(table, 4)
insert(table, 13)
print table
```

Select one:

a.
```
[0, 0, 13, 3, 4, 0, 0, 0, 0, 0]
```

b.
```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

c.
```
[0, 0, 0, 3, 4, 0, 0, 13, 0, 0]
```

✓

d.
```
[0, 0, 0, 13, 4, 0, 0, 0, 0, 0]
```

Your answer is correct.

The correct answer is:

```
[0, 0, 0, 3, 4, 0, 0, 13, 0, 0]
```