

Decimal to Binary Using Stacks

1. Express -15 as a 6-bit signed binary number.

- a) 001111
- b) 101111
- c) 101110
- d) 001110

Answer: b

2. Which of the following code snippet is used to convert decimal to binary numbers?

a)

```
public void convertBinary(int num)
{
    int bin[] = new int[50];
    int index = 0;
    while(num > 0)
    {
        bin[index++] = num%2;
        num = num/2;
    }
    for(int i = index-1;i >= 0;i--)
    {
        System.out.print(bin[i]);
    }
}
```

b)

```
public void convertBinary(int num)
{
    int bin[] = new int[50];
    int index = 0;
    while(num > 0)
    {
        bin[++index] = num%2;
        num = num/2;
    }
    for(int i = index-1;i >= 0;i--)
    {
        System.out.print(bin[i]);
    }
}
```

c)

```
public void convertBinary(int num)
{
    int bin[] = new int[50];
    int index = 0;
    while(num > 0)
    {
        bin[index++] = num/2;
        num = num%2;
    }
    for(int i = index-1;i >= 0;i--)
```

```

        {
            System.out.print(bin[i]);
        }
    }
d)
public void convertBinary(int num)
{
    int bin[] = new int[50];
    int index = 0;
    while(num > 0)
    {
        bin[++index] = num/2;
        num = num%2;
    }
    for(int i = index-1;i >= 0;i--)
    {
        System.out.print(bin[i]);
    }
}

```

Answer:

```

public void convertBinary(int num)
{
    int bin[] = new int[50];
    int index = 0;
    while(num > 0)
    {
        bin[index++] = num%2;
        num = num/2;
    }
    for(int i = index-1;i >= 0;i--)
    {
        System.out.print(bin[i]);
    }
}

```

3. Which is the predefined method available in Java to convert decimal to binary numbers?

- a) toBinaryInteger(int)
- b) toBinaryValue(int)
- c) toBinaryNumber(int)
- d) toBinaryString(int)

Answer: toBinaryString(int)

4. Using stacks, how to obtain the binary representation of the number?

```

a)
public void convertBinary(int num)
{
    Stack<Integer> stack = new Stack<Integer>();
    while (num != 0)
    {
        int digit = num / 2;
        stack.push(digit);
        num = num % 2;
    }
}

```

```

        System.out.print("\nBinary representation is:");
        while (!(stack.isEmpty() ))
        {
            System.out.print(stack.pop());
        }
    }
}
b)
public void convertBinary(int num)
{
    Stack<Integer> stack = new Stack<Integer>();
    while (num != 0)
    {
        int digit = num % 2;
        stack.push(digit);
    }
    System.out.print("\nBinary representation is:");
    while (!(stack.isEmpty() ))
    {
        System.out.print(stack.pop());
    }
}
c)
public void convertBinary(int num)
{
    Stack<Integer> stack = new Stack<Integer>();
    while (num != 0)
    {
        int digit = num % 2;
        stack.push(digit);
        num = num / 2;
    }
    System.out.print("\nBinary representation is:");
    while (!(stack.isEmpty() ))
    {
        System.out.print(stack.pop());
    }
}
d)
public void convertBinary(int num)
{
    Stack<Integer> stack = new Stack<Integer>();
    while (num != 0)
    {
        int digit = num % 2;
        stack.push(digit%2);
        num = num / 2;
    }
    System.out.print("\nBinary representation is:");
    while (!(stack.isEmpty() ))
    {
        System.out.print(stack.pop());
    }
}
}

```

Answer:

```
public void convertBinary(int num)
{
    Stack<Integer> stack = new Stack<Integer>();
    while (num != 0)
    {
        int digit = num % 2;
        stack.push(digit);
        num = num / 2;
    }
    System.out.print("nBinary representation is:");
    while (!(stack.isEmpty() ))
    {
        System.out.print(stack.pop());
    }
}
```

5. What is the time complexity for converting decimal to binary numbers?

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n \log n)$

Answer: $O(\log n)$

6. Write a piece of code which returns true if the string contains balanced parenthesis, false otherwise.

```
a)
public boolean isBalanced(String exp)
{
    int len = exp.length();
    Stack<Integer> stk = new Stack<Integer>();
    for(int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            if(stk.peek() == null)
            {
                return false;
            }
            stk.pop();
        }
    }
    return true;
}
```

```
b)
public boolean isBalanced(String exp)
{
    int len = exp.length();
    Stack<Integer> stk = new Stack<Integer>();
    for(int i = 0; i < len; i++)
    {
```

```

        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            if(stk.peek() != null)
            {
                return true;
            }
            stk.pop();
        }
    }
    return false;
}
}
c)
public boolean isBalanced(String exp)
{
    int len = exp.length();
    Stack<Integer> stk = new Stack<Integer>();
    for(int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == ')')
            stk.push(i);
        else if (ch == '(')
        {
            if(stk.peek() == null)
            {
                return false;
            }
            stk.pop();
        }
    }
    return true;
}
}
d)
public boolean isBalanced(String exp)
{
    int len = exp.length();
    Stack<Integer> stk = new Stack<Integer>();
    for(int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            if(stk.peek() != null)
            {
                return false;
            }
            stk.pop();
        }
    }
}
}

```

```

    }
    return true;
}

```

Answer:

```

public boolean isBalanced(String exp)
{
    int len = exp.length();
    Stack<Integer> stk = new Stack<Integer>();
    for(int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            if(stk.peek() == null)
            {
                return false;
            }
            stk.pop();
        }
    }
    return true;
}

```

7. What is the time complexity of the following code?

```

public boolean isBalanced(String exp)
{
    int len = exp.length();
    Stack<Integer> stk = new Stack<Integer>();
    for(int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            if(stk.peek() == null)
            {
                return false;
            }
            stk.pop();
        }
    }
    return true;
}

```

- a) $O(\log n)$
- b) $O(n)$
- c) $O(1)$
- d) $O(n \log n)$

Answer: $O(n)$

8. Which of the following program prints the index of every matching parenthesis?

- a)

```

public void dispIndex(String exp)
{
    Stack<Integer> stk = new Stack<Integer>();
    for (int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            try
            {
                int p = stk.pop() + 1;
                System.out.println("'')' at index "+(i+1)+" matched with '('' at
index "+p);
            }
            catch(Exception e)
            {
                System.out.println("'')' at index "+(i+1)+" is unmatched");
            }
        }
    }
    while (!stk.isEmpty() )
        System.out.println("'(' at index "+(stk.pop() +1)+" is unmatched");
}

```

```

b)
public void dispIndex(String exp)
{
    Stack<Integer> stk = new Stack<Integer>();
    for (int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            try
            {
                int p = stk.pop() + 1;
                System.out.println("'')' at index "+(i)+" matched with '('' at
index "+p);
            }
            catch(Exception e)
            {
                System.out.println("'')' at index "+(i)+" is unmatched");
            }
        }
    }
    while (!stk.isEmpty() )
        System.out.println("'(' at index "+(stk.pop() +1)+" is unmatched");
}

```

```

c)
public void dispIndex(String exp)

```

```

{
    Stack<Integer> stk = new Stack<Integer>();
    for (int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == ')')
            stk.push(i);
        else if (ch == '(')
        {
            try
            {
                int p = stk.pop() + 1;
                System.out.println("'" + ch + "' at index " + (i+1) + " matched with '" + ch + "' at
index " + p);
            }
            catch (Exception e)
            {
                System.out.println("'" + ch + "' at index " + (i+1) + " is unmatched");
            }
        }
    }
    while (!stk.isEmpty() )
        System.out.println("'" + ch + "' at index " + (stk.pop() + 1) + " is unmatched");
}
d)
public void dispIndex(String exp)
{
    Stack<Integer> stk = new Stack<Integer>();
    for (int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == ')')
            stk.push(i);
        else if (ch == '(')
        {
            try
            {
                int p = stk.pop();
                System.out.println("'" + ch + "' at index " + (i+1) + " matched with '" + ch + "' at
index " + p);
            }
            catch (Exception e)
            {
                System.out.println("'" + ch + "' at index " + (i+1) + " is unmatched");
            }
        }
    }
    while (!stk.isEmpty() )
        System.out.println("'" + ch + "' at index " + (stk.pop() + 1) + " is unmatched");
}

```

Answer:

```

public void dispIndex(String exp)
{

```



```
Stack<Integer> stk = new Stack<Integer>();
for (int i = 0; i < len; i++)
{
    char ch = exp.charAt(i);
    if (ch == '(')
        stk.push(i);
    else if (ch == ')')
    {
        try
        {
            int p = stk.pop() + 1;
            System.out.println("'')' at index "+(i+1)+" matched with ')' at
index "+p);
        }
        catch(Exception e)
        {
            System.out.println("'')' at index "+(i+1)+" is unmatched");
        }
    }
}
while (!stk.isEmpty() )
    System.out.println("'(' at index "+(stk.pop() +1)+" is unmatched");
}
```