
Question 1

1. What function would you use to join an element on the front of a list? (1 mark)
2. Assuming `l` is long enough, what expression involving `hd` and `tl` would give the 2nd element of the list `l`? (1 mark)
3. What expression involving the functions `nil` and `cons` could be used to construct the list `[1,2,3]`? (1 mark)

Please write your answers here:

Question 2

If `l1` has length 3, how long is the list returned by calling `append1(l1,l1)`? (Note that this is `append1` not `append`.)

☐ 3

☐ 0

☐ 4

☐ 1

☐ 6

Question 3

Look at the following code:

```
String f(String s) {  
    String to_return = "";  
    while(true) {  
        if(s.equals("")) return to_return;  
        // at this point s is not ""  
        char c2 = s.charAt(0);  
        if(c2< 'l') { to_return += c2; }  
        s=s.substring(1);  
    }  
}
```

What is the result of calling `f("hello")`?

- ☐ true
- ☐ false
- ☐ he
- ☐ helo
- ☐ hello
- ☐ ll

Question 4

Look at the following code:

```
List g(int i, List l) {  
    List to_return = nil();  
    while(true) {  
        if(l.isEmpty()) return to_return;  
        int j = (Integer)(hd(l));  
        if(i < j) {  
            to_return=append1(to_return,i);  
            to_return=append1(to_return,j);  
        }  
        i=j;  
        l=tl(l);  
    }  
}
```

What is the result of calling `g(3,[1,2,4,5])`?

- ☐ [1,2,4,5]
- ☐ [1,2,3]
- ☐ [1,2,2,4,4,5]
- ☐ [1,2,4,5,3]
- ☐ something else

Question 5

Look at the following code:

```
List r(int i0) {  
    List to_return = nil();  
    int i = i0;  
    while(true) {  
        if (i<=0) return to_return;  
        to_return = cons(0,to_return);  
        i=i-1;  
    }  
}
```

What is the result of calling `r(3)` ?

- ☐ [0,1,2]
 - ☐ [0,1,2,3]
 - ☐ [2,1,0]
 - ☐ [3,2,1,0]
 - ☐ [0,0,0]
-

Question 6

Look at the following code:

```
public String mn(String s) {  
    String to_return="";  
    String str = "";  
    while(true) {  
        if(s.equals("")) return to_return;  
        char c = s.charAt(0);  
        str=str+c;  
        to_return=to_return+str;  
        s=s.substring(1);  
    }  
}
```

What is the result of calling `mn("abc d")`?

- ☐ "abc d"
- ☐ "aababcabcbcd"
- ☐ "ababcabc abc d"
- ☐ "aababcabc abc d"
- ☐ something else

Question 7

Look at the following code:

```
boolean pn(String s1, String s2) {  
    while(true) {  
        if(s1.equals("")) return false;  
        if(s2.equals("")) return true;  
        if(s1.charAt(0)>s2.charAt(0)) return false;  
        s1=s1.substring(1);  
        s2=s2.substring(1);  
    }  
}
```

What is the result of calling `pn("tomab","tomc")`?

- ☐ An exception is thrown
 - ☐ The program fails to compile
 - ☐ true
 - ☐ false
 - ☐ something else
-

Question 8

Look at the following code:

```
String rs(String s, String to_return) {  
    if(s.equals("")) return to_return;  
    char c = s.charAt(0);  
    to_return = (c + to_return);  
    s=s.substring(1);  
    return rs(s,to_return);  
}
```

What is the result of calling `rs(rs("def","abc"),"")`?

- ☐ "abc"
- ☐ "defabc"
- ☐ "fedabc"
- ☐ "cbafed"
- ☐ something else

Question 9

Look at the following code:

```
public List uv(List l, int n) {  
    List to_return = nil();  
    while(true) {  
        if(l.isEmpty()) return to_return;  
        int i = (Integer)hd(l);  
        if(i*i<n) { to_return = append1(to_return,i); }  
        l=tl(l);  
    }  
}
```

What is the result of calling `uv([1,9,2,8,4,3],16)`?

- ☐ [1,2,3,4]
 - ☐ [1,2,3,4,8]
 - ☐ [1,2,3,4,8,9]
 - ☐ [1,2,3]
 - ☐ something else
-

Question 10

Look at the following code:

```
List wx(List l1, List l2) {  
    List to_return = nil();  
    while(true) {  
        if(l1.isEmpty()) return append(to_return,l2);  
        to_return = append1(to_return,hd(l1));  
        to_return = append1(to_return,hd(l2));  
        l1 = tl(l1);  
        l2 = tl(l2);  
    }  
}
```

What is the result of calling `wx([1,3,5,0,-2],[6,4,2])`?

- ☐ [1, 6, 3, 4, 5, 2, 0, -2]
- ☐ [1, 6, 3, 4, 5, 2]
- ☐ [1, 3, 5, 6, 4, 2, 0, -2]
- ☐ [-2, 0, 1, 2, 3, 4, 5, 6]
- ☐ something else