**Question C – Daniyar Nazarbayev, H00204990.**

1.

fun drop (x:int, l:int list) = if x=0 then l else drop(x-1,tl(l));

2.

fun take (x:int, l:string list) = if x=0 then [] else hd(l)::take(x-1,tl(l));

3.

fun starts (l1:int list, l2:int list) = if (l1<>[] andalso l2<>[] andalso hd(l1)=hd(l2))

then (true andalso starts(tl(l1),tl(l2)))

else if (l1<>[] andalso l2<>[] andalso hd(l1)<>hd(l2)) then false

else if (l1=[] andalso l2<>[]) then true

else if (l2=[] andalso l1<>[]) then false

else true;

4.

fun contains (l1:string list, l2:string list) =

let val x = l1;

in

if l1<>[] andalso l2<>[] andalso hd(l1)=hd(l2)

then true andalso contains (tl(l1), tl(l2))

else if l1<>[] andalso l2<>[] andalso hd(l1)<>hd(l2)

then false orelse contains (x, tl(l2))

else if l1=[]

then true

else false

end;

5.

*fun contains (l1:int list, l2:int list) =*

*let val x = l1;*

*in*

*if l1<>[] andalso l2<>[] andalso hd(l1)=hd(l2)*

*then true andalso contains (tl(l1), tl(l2))*

*else if l1<>[] andalso l2<>[] andalso hd(l1)<>hd(l2)*

*then false orelse contains (x, tl(l2))*

*else if l1=[]*

*then true*

*else false*

*end;*

fun delete (l1:int list, l2:int list) =

let

val y = l1;

val z = l2;

in

if contains(y,z)<>true

then l2

else if l1<>[] andalso l2<>[] andalso contains(l1, List.take(l2, length(l1)))

then delete (tl(l1), tl(l2))

else if l1<>[] andalso l2<>[]

then hd(l2)::delete (y, tl(l2))

else if l1=[]

then l2

else if l2=[]

then []

else []

end;

6.

**val a: int list = [3,2];**

val w: int list = [3,2,1,2,3,2,1,2,3];

*fun contains (l1:int list, l2:int list) =*

*let val x = l1;*

*in*

*if l1<>[] andalso l2<>[] andalso hd(l1)=hd(l2)*

*then true andalso contains (tl(l1), tl(l2))*

*else if l1<>[] andalso l2<>[] andalso hd(l1)<>hd(l2)*

*then false orelse contains (x, tl(l2))*

*else if l1=[]*

*then true*

*else false*

*end;*

fun deleteAll(l1:int list, l2:int list) =

if contains(l1,l2)<>true

then l2

else if l1<>[] andalso l2<>[] andalso contains(l1, List.take(l2, length(l1)))

then deleteAll(tl(l1), tl(l2))

else if l1<>[] andalso l2<>[]

then hd(l2)::deleteAll(a, tl(l2))

else if l1=[] andalso l2<>[]

then deleteAll(a, l2)

else [];

unfortunately I cannot use *let in end*, since the value of the first list will change after it finds it’s first list pattern to []. Because of that initialize the list you want beforehand out of the loop and call it a.