SOLUTIONS - HW 3

```
exception EverythingDivide0 of string;

fun ListDivisors1 (n,m) =
if m=1
then [1]
else
  if n mod m = 0
  then m::ListDivisors1(n,m-1)
  else ListDivisors1(n,m-1);

val ListDivisors1 = fn : int * int -> int list

fun ListDivisors (n) =
if n = 0
then raise EverythingDivide0 "Everything divide 0"
else ListDivisors1(n,n);

val ListDivisors = fn : int -> int list

ListDivisors(0);
uncaught exception EverythingDivide0

ListDivisors(1);
val it = [1] : int list

ListDivisors(2);
val it = [2,1] : int list

ListDivisors(100);
val it = [100,50,25,20,10,5,4,2,1] : int list

ListDivisors(6);
val it = [6,3,2,1] : int list

ListDivisors(28);
val it = [28,14,7,4,2,1] : int list

ListDivisors(496);
val it = [496,248,124,62,31,16,8,4,2,1] : int list

ListDivisors(8128);
val it = [8128,4064,2032,1016,508,254,127,64,32,16,8,4,...] : int list
```

First method: (without using ListDivisors(n))

```
fun SumDivisors1 (n,m) =
if m=1
then 1
else
  if n mod m = 0
  then m + SumDivisors1(n,m-1)
  else SumDivisors1(n,m-1);

val SumDivisors1 = fn : int * int -> int

fun SumDivisors n =
if n = 0
then raise EverythingDivide0 "Everything divide 0"
else SumDivisors1(n,n);

val SumDivisors = fn : int -> int

SumDivisors(0);
uncaught exception EverythingDivide0

SumDivisors(1);
val it = 1 : int

SumDivisors(6);
val it = 12 : int

SumDivisors(12);
val it = 28 : int

SumDivisors(100);
val it = 217 : int

SumDivisors(28);
val it = 56 : int
int
SumDivisors(496);
val it = 992 : int

SumDivisors(8128);
val it = 16256 : int
```

Second method: (using ListDivisors(n))

SumList: sum of the elements of a list.

```
fun SumList([]) = 0
| SumList(x::L) = x + SumList(L);

val SumList = fn : int list -> int

SumList([1,2]);
val it = 3 : int

SumList([1,3,4]);
val it = 8 : int

SumList([]);
val it = 0 : int

fun SumDivisors(n) =
let val L = ListDivisors(n)
in
SumList(L)
end;

val SumDivisors = fn : int -> int

SumDivisors(0);
uncaught exception LDEverything

SumDivisors(1);
val it = 1 : int

SumDivisors(2);
val it = 3 : int

umDivisors(6);
val it = 12 : int

SumDivisors(28);
val it = 56 : int

SumDivisors(496);
val it = 992 : int

SumDivisors(8128);
val it = 16256 : int

fun IsPerfect(n) =
```

```
if n=1 then true
else
if SumDivisors(n) = 2*n
then true
else false;

IsPerfect(6);
val it = true : bool

IsPerfect(28);
val it = true : bool

IsPerfect(150);
val it = false : bool

IsPerfect(496);
val it = true : bool

IsPerfect(8128);
val it = true : bool

IsPerfect(495);
val it = false : bool

ListPerfect: In decreasing order.

fun ListPerfect(n) =
if n < 1 then []
else if IsPerfect(n)
    then n::ListPerfect(n-1)
     else ListPerfect(n-1);

val ListPerfect = fn : int -> int list

ListPerfect(29);
val it = [28,6] : int list

ListPerfect(400);
val it = [28,6] : int list

ListPerfect(496);
val it = [496,28,6] : int list

ListPerfect(600);
val it = [496,28,6] : int list
```

```
ListPerfect(8129);
val it = [8128,496,28,6] : int list
```