- Introduction and overview
- Basic types, definitions and functions
- Basic data structures
- More advanced data structures
- Higher order functions
- **Exceptions, input/output and imperative constructs**

Table of Contents

Imperative features in OCaml

Getting and handling your Exceptions
Week 5 Echéance le déc 12, 2016 at 23:30 UTC

Getting information in and out
Week 5 Echéance le déc 12, 2016 at 23:30 UTC

Sequences and iterations
Week 5 Echéance le déc 12, 2016 at 23:30 UTC

Mutable arrays
Week 5 Echéance le déc 12, 2016 at 23:30 UTC

Mutable record fields
Week 5 Echéance le déc 12, 2016 at 23:30 UTC

Variables, aka References
Week 5 Echéance le déc 12, 2016 at 23:30 UTC

- Modules and data abstraction
- Project

IMPLEMENTING MUTABLE LISTS (80/80 points)

Using mutable record fields, we can define the type of a list data structure with explicit pointers, as defined by the type `'a xlist` given in the prelude.

The empty list is written:

```
{ pointer = Nil }
```

The singleton list containing only `"one"` is written:

```
{ pointer = List (1, { pointer = Nil }) }
```

The list containing the elements `1`, then `2` then `3` is written:

```
{ pointer =
    List (1, { pointer =
                 List (2, { pointer =
                              List (3, { pointer =
                                           Nil }) }) }) }
```

1. Define `head : 'a xlist -> 'a` that returns the first element of the list if it exists, or fails with `Empty_xlist`.
   This function does not modify the list.
2. Define `tail : 'a xlist -> 'a xlist` that returns the list without its first element if it exists, or fails with `Empty_xlist`.
   This function does not modify the list.
3. Define `add : 'a -> 'a xlist -> unit` that modifies the list in place to add an element at the front.
4. Define `chop : 'a xlist -> unit` that modifies the list to remove its front element, or fails with `Empty_xlist`.
5. Define `append : 'a xlist -> 'a xlist -> unit`, a destructive concatenation operation that modifies the last pointer of the first list to point to the beginning of the second list.
6. Define `filter : ('a -> bool) -> 'a xlist -> unit`, a destructive filter operation on lists that removes from the list all elements that do not satisfy the boolean predicate passed as parameter.

THE GIVEN PRELUDE

```
type 'a xlist =
  { mutable pointer : 'a cell }
and 'a cell =
  | Nil
  | List of 'a * 'a xlist ;;

let nil () =
  { pointer = Nil } ;;

let cons elt rest =
  { pointer = List (elt, rest) } ;;

exception Empty_xlist ;;
```

YOUR OCAML ENVIRONMENT

```
let tail l =
  match l.pointer with
  | Nil -> raise Empty_xlist
  | List (a, list) -> list
;;

let add a l =
  let rec ajout elt liste =
    match liste.pointer with
    | Nil -> { pointer = List (elt, nil ())}
    | List (e, tail) ->  { pointer = List (elt, (ajout e tail))}
  in l.pointer <- (ajout a l).pointer
;;

let chop l =
  match l.pointer with
  | Nil -> raise Empty_xlist
  | List (a, list) -> l.pointer <- list.pointer
;;

let append l l' =
  let rec ajout liste1 liste2=
    match liste1.pointer with
    | Nil -> { pointer = liste2.pointer}
    | List (a, tail) ->  { pointer = List (a, (ajout tail liste2))}
  in l.pointer <- (ajout l l').pointer
;;
```

Switch >>

Typecheck

Reset Template

Full-screen [+]

Check & Save

Exercise complete (click for details) 80 pts

v Exercise 1: head Completed, 20 pts

Found head with compatible type.

```
Computing
  head
    {pointer =
      List (0,
       {pointer =
         List (2,
          {pointer =
            List (4,
             {pointer =
               List (-3,
                {pointer =
                  List (3,
                   {pointer = List (-1, {pointer = List (3, {pointer = Nil})})})})})})})}
```

Correct value 0 1 pt

Correct value 1 pt

```
  {pointer =
    List (0,
     {pointer =
       List (2,
        {pointer =
          List (4,
           {pointer =
             List (-3,
              {pointer =
                List (3,
                 {pointer = List (-1, {pointer = List (3, {pointer = Nil})})})})})})})}
```

```
Computing
  head
    {pointer =
      List (2, {pointer = List (-1, {pointer = List (3, {pointer = Nil})})})}
```

Correct value 2 1 pt

Correct value 1 pt

```
  {pointer =
    List (2, {pointer = List (-1, {pointer = List (3, {pointer = Nil})})})}
```

```
Computing
  head
    {pointer =
      List (0,
       {pointer =
         List (2,
          {pointer =
            List (-4,
             {pointer =
               List (0,
                {pointer =
                  List (-4,
                   {pointer = List (3, {pointer = List (3, {pointer = Nil})})})})})})})}
```

Correct value 0 1 pt

Correct value 1 pt

```
  {pointer =
    List (0,
     {pointer =
       List (2,
        {pointer =
          List (-4,
           {pointer =
             List (0,
Computing head {pointer = List (1, {pointer = List (-3, {pointer = Nil})})}
Correct value 1                                                              1 pt
Correct value {pointer = List (1, {pointer = List (-3, {pointer = Nil})})}   1 pt
Computing
  head
    {pointer =
      List (0, {pointer = List (3, {pointer = List (4, {pointer = Nil})})})}
Correct value 0                                                              1 pt
Correct value                                                                1 pt
  {pointer =
    List (0, {pointer = List (3, {pointer = List (4, {pointer = Nil})})})}
Computing
  head
    {pointer =
      List (1,
       {pointer =
         List (4,
          {pointer =
            List (-3,
             {pointer =
               List (-2,
                {pointer =
                  List (-3,
                   {pointer = List (-2, {pointer = List (1, {pointer = Nil})})})})})})})}
Correct value 1                                                              1 pt
Correct value                                                                1 pt
  {pointer =
    List (1,
     {pointer =
       List (4,
        {pointer =
          List (-3,
           {pointer =
             List (-2,
              {pointer =
                List (-3,
                 {pointer = List (-2, {pointer = List (1, {pointer = Nil})})})})})})})}
Computing head {pointer = Nil}
Correct exception Empty_xlist                                                1 pt
Correct exception Empty_xlist                                                1 pt
Computing
  head
    {pointer =
      List (-2,
       {pointer =
         List (-1,
          {pointer =
            List (3,
             {pointer =
               List (0,
                {pointer = List (-2, {pointer = List (-3, {pointer = Nil})})})})})})}
Correct value -2                                                             1 pt
Correct value                                                                1 pt
  {pointer =
    List (-2,
     {pointer =
       List (-1,
        {pointer =
          List (3,
           {pointer =
             List (0,
              {pointer = List (-2, {pointer = List (-3, {pointer = Nil})})})})})})}
Computing
  head
    {pointer =
      List (-5,
       {pointer =
         List (-1,
          {pointer = List (-3, {pointer = List (1, {pointer = Nil})})})})}
Correct value -5                                                             1 pt
Correct value                                                                1 pt
  {pointer =
    List (-5,
     {pointer =
       List (-1, {pointer = List (-3, {pointer = List (1, {pointer = Nil})})})})}
Computing head {pointer = Nil}
Correct exception Empty_xlist                                                1 pt
Correct exception Empty_xlist                                                1 pt
v Exercise 2: tail                                            Completed, 20 pts
Found tail with compatible type.
Computing
  tail
    {pointer =
      List (-4,
       {pointer =
             {pointer =
               List (-5,
                {pointer =
                  List (-1,
                   {pointer =
                     List (-4,
                      {pointer =
                        List (2,
                         {pointer =
                           List (0, {pointer = List (-3, {pointer = Nil})})})})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List (-2,
   {pointer =
     List (1,
      {pointer =
        List (-5,
         {pointer =
           List (-1,
            {pointer =
              List (-4,
               {pointer =
                 List (2,
                  {pointer = List (0, {pointer = List (-3, {pointer = Nil})})})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List (-4,
   {pointer =
     List (-2,
      {pointer =
        List (1,
         {pointer =
           List (-5,
            {pointer =
              List (-1,
               {pointer =
                 List (-4,
                  {pointer =
                    List (2,
                     {pointer =
                       List (0, {pointer = List (-3, {pointer = Nil})})})})})})})})})}
```

Computing

```
tail
  {pointer =
    List (3,
     {pointer =
       List (2,
        {pointer =
          List (4,
           {pointer =
             List (4,
              {pointer =
                List (-1,
                 {pointer =
                   List (2,
                    {pointer =
                      List (3,
                       {pointer =
                         List (-5,
                          {pointer =
                            List (1, {pointer = List (0, {pointer = Nil})})})})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List (2,
   {pointer =
     List (4,
      {pointer =
        List (4,
         {pointer =
           List (-1,
            {pointer =
              List (2,
               {pointer =
                 List (3,
                  {pointer =
                    List (-5,
                     {pointer =
                       List (1, {pointer = List (0, {pointer = Nil})})})})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List (3,
   {pointer =
     List (2,
      {pointer =
        List (4,
         {pointer =
           List (4,
            {pointer =
              List (-1,
               {pointer =
                 List (2,
                          {pointer =
                            List (1, {pointer = List (0, {pointer = Nil})})})})})})})})})})}
Computing
  tail
    {pointer =
      List (-1,
       {pointer =
         List (0,
          {pointer =
            List (2,
             {pointer =
               List (-4,
                {pointer =
                  List (4,
                   {pointer =
                     List (0,
                      {pointer =
                        List (-3,
                         {pointer =
                           List (4,
                            {pointer =
                              List (4,
                               {pointer =
                                 List (-5,
                                  {pointer = List (-5, {pointer = Nil})})})})})})})})})})})}
Correct value                                                                    1 pt
  {pointer =
    List (0,
     {pointer =
       List (2,
        {pointer =
          List (-4,
           {pointer =
             List (4,
              {pointer =
                List (0,
                 {pointer =
                   List (-3,
                    {pointer =
                      List (4,
                       {pointer =
                         List (4,
                          {pointer =
                            List (-5, {pointer = List (-5, {pointer = Nil})})})})})})})})})}
Correct value                                                                    1 pt
  {pointer =
    List (-1,
     {pointer =
       List (0,
        {pointer =
          List (2,
           {pointer =
             List (-4,
              {pointer =
                List (4,
                 {pointer =
                   List (0,
                    {pointer =
                      List (-3,
                       {pointer =
                         List (4,
                          {pointer =
                            List (4,
                             {pointer =
                               List (-5,
                                {pointer = List (-5, {pointer = Nil})})})})})})})})})})})}
Computing tail {pointer = Nil}
Correct exception Empty_xlist                                                    1 pt
Correct exception Empty_xlist                                                    1 pt
Computing
  tail
    {pointer =
      List (-1, {pointer = List (1, {pointer = List (4, {pointer = Nil})})})}
Correct value {pointer = List (1, {pointer = List (4, {pointer = Nil})})}        1 pt
Correct value                                                                    1 pt
  {pointer =
    List (-1, {pointer = List (1, {pointer = List (4, {pointer = Nil})})})}
Computing
  tail
    {pointer =
      List (-5,
       {pointer =
         List (2,
          {pointer =
            List (2,
             {pointer =
               List (0,
                {pointer =
                    {pointer =
                      List (-2, {pointer = List (3, {pointer = Nil})})})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List (2,
   {pointer =
     List (2,
      {pointer =
        List (0,
         {pointer =
           List (1,
            {pointer =
              List (0,
               {pointer = List (-2, {pointer = List (3, {pointer = Nil})})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List (-5,
   {pointer =
     List (2,
      {pointer =
        List (2,
         {pointer =
           List (0,
            {pointer =
              List (1,
               {pointer =
                 List (0,
                  {pointer = List (-2, {pointer = List (3, {pointer = Nil})})})})})})})})}
```

Computing

```
tail
  {pointer =
    List (-5,
     {pointer =
       List (4,
        {pointer =
          List (2,
           {pointer =
             List (4,
              {pointer =
                List (-5,
                 {pointer =
                   List (-1,
                    {pointer =
                      List (3,
                       {pointer =
                         List (1,
                          {pointer =
                            List (-2,
                             {pointer =
                               List (-3,
                                {pointer = List (1, {pointer = Nil})})})})})})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List (4,
   {pointer =
     List (2,
      {pointer =
        List (4,
         {pointer =
           List (-5,
            {pointer =
              List (-1,
               {pointer =
                 List (3,
                  {pointer =
                    List (1,
                     {pointer =
                       List (-2,
                        {pointer =
                          List (-3, {pointer = List (1, {pointer = Nil})})})})})})})})})})
```

Correct value 1 pt

```
{pointer =
  List (-5,
   {pointer =
     List (4,
      {pointer =
        List (2,
         {pointer =
           List (4,
            {pointer =
              List (-5,
               {pointer =
                 List (-1,
                  {pointer =
                    List (3,
                     {pointer =
                       List (1,
                        {pointer =
                          List (-2,
                           {pointer =
                             List (-3, {pointer = List (1, {pointer = Nil})})})})})})})})}
```

```
       {pointer =
         List (-5,
          {pointer = List (0, {pointer = List (-3, {pointer = Nil})})})})}
Correct value                                                                1 pt
  {pointer =
    List (-5, {pointer = List (0, {pointer = List (-3, {pointer = Nil})})})}
Correct value                                                                1 pt
  {pointer =
    List (2,
     {pointer =
       List (-5, {pointer = List (0, {pointer = List (-3, {pointer = Nil})})})})}
Computing tail {pointer = List (4, {pointer = List (4, {pointer = Nil})})}
Correct value {pointer = List (4, {pointer = Nil})}                          1 pt
Correct value {pointer = List (4, {pointer = List (4, {pointer = Nil})})}    1 pt
Computing
  tail
    {pointer =
      List (-3,
       {pointer =
         List (3,
          {pointer =
            List (2,
             {pointer =
               List (2,
                {pointer =
                  List (-5,
                   {pointer = List (-5, {pointer = List (0, {pointer = Nil})})})})})})})}
Correct value                                                                1 pt
  {pointer =
    List (3,
     {pointer =
       List (2,
        {pointer =
          List (2,
           {pointer =
             List (-5,
              {pointer = List (-5, {pointer = List (0, {pointer = Nil})})})})})})}
Correct value                                                                1 pt
  {pointer =
    List (-3,
     {pointer =
       List (3,
        {pointer =
          List (2,
           {pointer =
             List (2,
              {pointer =
                List (-5,
                 {pointer = List (-5, {pointer = List (0, {pointer = Nil})})})})})})})}
```

v Exercise 3: add Completed, 10 pts

```
Found add with compatible type.
Computing
  add
    'g'
    {pointer =
      List ('f',
       {pointer =
         List ('i',
          {pointer =
            List ('l',
             {pointer =
               List ('o',
                {pointer =
                  List ('o',
                   {pointer =
                     List ('m', {pointer = List ('r', {pointer = Nil})})})})})})})}
Correct value                                                                1 pt
  {pointer =
    List ('g',
     {pointer =
       List ('f',
        {pointer =
          List ('i',
           {pointer =
             List ('l',
              {pointer =
                List ('o',
                 {pointer =
                   List ('o',
                    {pointer =
                      List ('m', {pointer = List ('r', {pointer = Nil})})})})})})})})}
Computing add 'e' {pointer = List ('c', {pointer = Nil})}
Correct value {pointer = List ('e', {pointer = List ('c', {pointer = Nil})})}    1 pt
Computing
  add
    'r'
    {pointer =
  {pointer =
    List ('r',
     {pointer =
       List ('p',
        {pointer = List ('t', {pointer = List ('g', {pointer = Nil})})})})}
Computing
  add
    'n'
    {pointer =
      List ('x',
       {pointer =
         List ('r',
          {pointer = List ('i', {pointer = List ('g', {pointer = Nil})})})})}
Correct value                                                                          1 pt
  {pointer =
    List ('n',
     {pointer =
       List ('x',
        {pointer =
          List ('r',
           {pointer = List ('i', {pointer = List ('g', {pointer = Nil})})})})})}
Computing
  add
    'b'
    {pointer =
      List ('u',
       {pointer =
         List ('x',
          {pointer =
            List ('i',
             {pointer =
               List ('z',
                {pointer = List ('k', {pointer = List ('t', {pointer = Nil})})})})})})}
Correct value                                                                          1 pt
  {pointer =
    List ('b',
     {pointer =
       List ('u',
        {pointer =
          List ('x',
           {pointer =
             List ('i',
              {pointer =
                List ('z',
                 {pointer = List ('k', {pointer = List ('t', {pointer = Nil})})})})})})})}
Computing
  add
    'f'
    {pointer =
      List ('q',
       {pointer =
         List ('h',
          {pointer =
            List ('c',
             {pointer =
               List ('x',
                {pointer =
                  List ('u',
                   {pointer =
                     List ('r', {pointer = List ('y', {pointer = Nil})})})})})})})}
Correct value                                                                          1 pt
  {pointer =
    List ('f',
     {pointer =
       List ('q',
        {pointer =
          List ('h',
           {pointer =
             List ('c',
              {pointer =
                List ('x',
                 {pointer =
                   List ('u',
                    {pointer =
                      List ('r', {pointer = List ('y', {pointer = Nil})})})})})})})})}
Computing add 'f' {pointer = List ('u', {pointer = List ('b', {pointer = Nil})})}
Correct value                                                                          1 pt
  {pointer =
    List ('f', {pointer = List ('u', {pointer = List ('b', {pointer = Nil})})})}
Computing
  add
    'x'
    {pointer =
      List ('x',
       {pointer =
         List ('x',
          {pointer =
            List ('x',
             {pointer =
               List ('s',
                {pointer =
                      {pointer =
                        List ('s',
                         {pointer =
                           List ('i',
                            {pointer =
                              List ('b',
                               {pointer =
                                 List ('p',
                                  {pointer = List ('j', {pointer = Nil})})})})})})})})})})})
Correct value                                                                        1 pt
  {pointer =
    List ('x',
     {pointer =
       List ('x',
        {pointer =
          List ('x',
           {pointer =
             List ('x',
              {pointer =
                List ('s',
                 {pointer =
                   List ('l',
                    {pointer =
                      List ('u',
                       {pointer =
                         List ('s',
                          {pointer =
                            List ('i',
                             {pointer =
                               List ('b',
                                {pointer =
                                  List ('p',
                                   {pointer = List ('j', {pointer = Nil})})})})})})})})})})})}
Computing
  add
    'g'
    {pointer =
      List ('c',
       {pointer =
         List ('p',
          {pointer =
            List ('j',
             {pointer =
               List ('w',
                {pointer =
                  List ('l',
                   {pointer =
                     List ('e',
                      {pointer =
                        List ('k',
                         {pointer =
                           List ('j', {pointer = List ('t', {pointer = Nil})})})})})})})})})
Correct value                                                                        1 pt
  {pointer =
    List ('g',
     {pointer =
       List ('c',
        {pointer =
          List ('p',
           {pointer =
             List ('j',
              {pointer =
                List ('w',
                 {pointer =
                   List ('l',
                    {pointer =
                      List ('e',
                       {pointer =
                         List ('k',
                          {pointer =
                            List ('j', {pointer = List ('t', {pointer = Nil})})})})})})})})}
Computing
  add
    'o'
    {pointer =
      List ('i',
       {pointer =
         List ('u',
          {pointer =
            List ('a',
             {pointer =
               List ('u',
                {pointer = List ('y', {pointer = List ('l', {pointer = Nil})})})})})})}
Correct value                                                                        1 pt
  {pointer =
    List ('o',
     {pointer =
       List ('i',
        {pointer =
          List ('u',
           {pointer =
```

v Exercise 4: chop Completed, 10 pts

Found chop with compatible type.

Computing
```
  chop
    {pointer =
      List (false,
       {pointer =
         List (false,
          {pointer =
            List (true,
             {pointer =
               List (false,
                {pointer =
                  List (false, {pointer = List (true, {pointer = Nil})})})})})})}
```
Correct value 1 pt
```
  {pointer =
    List (false,
     {pointer =
       List (true,
        {pointer =
          List (false,
           {pointer = List (false, {pointer = List (true, {pointer = Nil})})})})})}
```
Computing
```
  chop
    {pointer =
      List (true,
       {pointer =
         List (false,
          {pointer =
            List (false,
             {pointer =
               List (true,
                {pointer =
                  List (false,
                   {pointer =
                     List (false,
                      {pointer =
                        List (true,
                         {pointer =
                           List (true,
                            {pointer =
                              List (true,
                               {pointer =
                                 List (true,
                                  {pointer = List (false, {pointer = Nil})})})})})})})})})})})})
```
Correct value 1 pt
```
  {pointer =
    List (false,
     {pointer =
       List (false,
        {pointer =
          List (true,
           {pointer =
             List (false,
              {pointer =
                List (false,
                 {pointer =
                   List (true,
                    {pointer =
                      List (true,
                       {pointer =
                         List (true,
                          {pointer =
                            List (true,
                             {pointer = List (false, {pointer = Nil})})})})})})})})})})})}
```
Computing
```
  chop
    {pointer =
      List (true,
       {pointer = List (true, {pointer = List (false, {pointer = Nil})})})}
```
Correct value `{pointer = List (true, {pointer = List (false, {pointer = Nil})})}` 1 pt

Computing
```
  chop
    {pointer =
      List (true,
       {pointer = List (false, {pointer = List (true, {pointer = Nil})})})}
```
Correct value `{pointer = List (false, {pointer = List (true, {pointer = Nil})})}` 1 pt

Computing
```
  chop
    {pointer =
      List (false,
       {pointer =
         List (true,
          {pointer =
            List (true,
             {pointer =
               List (false,
                {pointer =
                      {pointer =
                        List (true, {pointer = List (true, {pointer = Nil})})})})})})})})}
```

Correct value 1 pt

```
  {pointer =
    List (true,
     {pointer =
       List (true,
        {pointer =
          List (false,
           {pointer =
             List (true,
              {pointer =
                List (true,
                 {pointer =
                   List (true, {pointer = List (true, {pointer = Nil})})})})})})})}
```

Computing

```
  chop
    {pointer =
      List (true,
       {pointer =
         List (true,
          {pointer =
            List (true,
             {pointer =
               List (true,
                {pointer =
                  List (false,
                   {pointer =
                     List (true,
                      {pointer =
                        List (false,
                         {pointer =
                           List (false,
                            {pointer = List (false, {pointer = Nil})})})})})})})})})}
```

Correct value 1 pt

```
  {pointer =
    List (true,
     {pointer =
       List (true,
        {pointer =
          List (true,
           {pointer =
             List (false,
              {pointer =
                List (true,
                 {pointer =
                   List (false,
                    {pointer =
                      List (false, {pointer = List (false, {pointer = Nil})})})})})})})})}
```

Computing

```
  chop
    {pointer =
      List (true,
       {pointer =
         List (false,
          {pointer =
            List (false,
             {pointer =
               List (false,
                {pointer =
                  List (false,
                   {pointer =
                     List (false, {pointer = List (false, {pointer = Nil})})})})})})})}
```

Correct value 1 pt

```
  {pointer =
    List (false,
     {pointer =
       List (false,
        {pointer =
          List (false,
           {pointer =
             List (false,
              {pointer =
                List (false, {pointer = List (false, {pointer = Nil})})})})})})}
```

Computing chop `{pointer = List (false, {pointer = List (false, {pointer = Nil})})}`

Correct value `{pointer = List (false, {pointer = Nil})}` 1 pt

Computing

```
  chop
    {pointer =
      List (true,
       {pointer = List (false, {pointer = List (false, {pointer = Nil})})})}
```

Correct value `{pointer = List (false, {pointer = List (false, {pointer = Nil})})}` 1 pt

Computing chop `{pointer = Nil}`

Correct exception `Empty_xlist` 1 pt

v Exercise 5: append Completed, 10 pts

Found append with compatible type.

Computing

```
       {pointer =
         List ('f',
          {pointer =
            List ('x',
             {pointer =
               List ('o',
                {pointer =
                  List ('o',
                   {pointer =
                     List ('t',
                      {pointer =
                        List ('z', {pointer = List ('s', {pointer = Nil})})})})})})})})}
Correct value                                                                        1 pt
  {pointer =
    List ('g',
     {pointer =
       List ('b',
        {pointer =
          List ('k',
           {pointer =
             List ('f',
              {pointer =
                List ('x',
                 {pointer =
                   List ('o',
                    {pointer =
                      List ('o',
                       {pointer =
                         List ('t',
                          {pointer =
                            List ('z', {pointer = List ('s', {pointer = Nil})})})})})})})})}
Computing
  append
    {pointer = List ('o', {pointer = List ('u', {pointer = Nil})})}
    {pointer = List ('i', {pointer = List ('g', {pointer = Nil})})}
Correct value                                                                        1 pt
  {pointer =
    List ('o',
     {pointer =
       List ('u',
        {pointer = List ('i', {pointer = List ('g', {pointer = Nil})})})})}
Computing
  append
    {pointer = Nil}
    {pointer =
      List ('q',
       {pointer = List ('z', {pointer = List ('s', {pointer = Nil})})})}
Correct value                                                                        1 pt
  {pointer =
    List ('q', {pointer = List ('z', {pointer = List ('s', {pointer = Nil})})})}
Computing
  append
    {pointer =
      List ('z',
       {pointer =
         List ('z',
          {pointer =
            List ('n',
             {pointer =
               List ('f',
                {pointer =
                  List ('d',
                   {pointer =
                     List ('s',
                      {pointer =
                        List ('z', {pointer = List ('o', {pointer = Nil})})})})})})})})}
    {pointer =
      List ('i',
       {pointer =
         List ('s',
          {pointer =
            List ('q',
             {pointer =
               List ('t',
                {pointer =
                  List ('a',
                   {pointer =
                     List ('g',
                      {pointer =
                        List ('q', {pointer = List ('q', {pointer = Nil})})})})})})})})}
Correct value                                                                        1 pt
  {pointer =
    List ('z',
     {pointer =
       List ('z',
        {pointer =
          List ('n',
           {pointer =
             List ('f',
              {pointer =
                    {pointer =
                      List ('z',
                       {pointer =
                         List ('o',
                          {pointer =
                            List ('i',
                             {pointer =
                               List ('s',
                                {pointer =
                                  List ('q',
                                   {pointer =
                                     List ('t',
                                      {pointer =
                                        List ('a',
                                         {pointer =
                                           List ('g',
                                            {pointer =
                                              List ('q',
                                               {pointer =
                                                 List ('q', {pointer = Nil})})})})})})})})})})
```

```
Computing
  append
    {pointer =
      List ('g',
       {pointer =
         List ('r',
          {pointer =
            List ('p',
             {pointer =
               List ('q',
                {pointer =
                  List ('s',
                   {pointer =
                     List ('l',
                      {pointer =
                        List ('t',
                         {pointer =
                           List ('k', {pointer = List ('c', {pointer = Nil})})})})})})})})})})
    {pointer =
      List ('c',
       {pointer =
         List ('p',
          {pointer =
            List ('l',
             {pointer =
               List ('a',
                {pointer =
                  List ('a',
                   {pointer =
                     List ('v',
                      {pointer =
                        List ('p',
                         {pointer =
                           List ('w', {pointer = List ('i', {pointer = Nil})})})})})})})})})})
```

Correct value 1 pt

```
  {pointer =
    List ('g',
     {pointer =
       List ('r',
        {pointer =
          List ('p',
           {pointer =
             List ('q',
              {pointer =
                List ('s',
                 {pointer =
                   List ('l',
                    {pointer =
                      List ('t',
                       {pointer =
                         List ('k',
                          {pointer =
                            List ('c',
                             {pointer =
                               List ('c',
                                {pointer =
                                  List ('p',
                                   {pointer =
                                     List ('l',
                                      {pointer =
                                        List ('a',
                                         {pointer =
                                           List ('a',
                                            {pointer =
                                              List ('v',
                                               {pointer =
                                                 List ('p',
                                                  {pointer =
                                                    List ('w',
```

Computing

```
append
  {pointer =
    List ('y',
     {pointer = List ('c', {pointer = List ('s', {pointer = Nil})})})}
  {pointer = Nil}
```

Correct value 1 pt

```
{pointer =
  List ('y', {pointer = List ('c', {pointer = List ('s', {pointer = Nil})})})}
```

Computing

```
append
  {pointer =
    List ('l',
     {pointer =
       List ('z',
        {pointer =
          List ('g',
           {pointer =
             List ('f',
              {pointer =
                List ('b',
                 {pointer =
                   List ('j', {pointer = List ('y', {pointer = Nil})})})})})})})}
  {pointer =
    List ('n',
     {pointer = List ('u', {pointer = List ('i', {pointer = Nil})})})}
```

Correct value 1 pt

```
{pointer =
  List ('l',
   {pointer =
     List ('z',
      {pointer =
        List ('g',
         {pointer =
           List ('f',
            {pointer =
              List ('b',
               {pointer =
                 List ('j',
                  {pointer =
                    List ('y',
                     {pointer =
                       List ('n',
                        {pointer =
                          List ('u', {pointer = List ('i', {pointer = Nil})})})})})})})})})}
```

Computing

```
append
  {pointer = List ('y', {pointer = List ('u', {pointer = Nil})})}
  {pointer = Nil}
```

Correct value `{pointer = List ('y', {pointer = List ('u', {pointer = Nil})})}` 1 pt

Computing

```
append
  {pointer =
    List ('a',
     {pointer =
       List ('w',
        {pointer =
          List ('m',
           {pointer =
             List ('x',
              {pointer =
                List ('m',
                 {pointer =
                   List ('x', {pointer = List ('h', {pointer = Nil})})})})})})})}
  {pointer =
    List ('m',
     {pointer =
       List ('x',
        {pointer =
          List ('w',
           {pointer =
             List ('z',
              {pointer = List ('p', {pointer = List ('c', {pointer = Nil})})})})})})}
```

Correct value 1 pt

```
{pointer =
  List ('a',
   {pointer =
     List ('w',
      {pointer =
        List ('m',
         {pointer =
           List ('x',
            {pointer =
              List ('m',
               {pointer =
                 List ('x',
                  {pointer =
                    List ('h',
                     {pointer =
                       List ('m',
                        {pointer =
                                  List ('z',
                                   {pointer =
                                     List ('p',
                                      {pointer = List ('c', {pointer = Nil})})})})})})})})})
```

Computing

```
  append
    {pointer =
      List ('s',
       {pointer =
         List ('v',
          {pointer =
            List ('n',
             {pointer =
               List ('l',
                {pointer =
                  List ('b',
                   {pointer =
                     List ('z', {pointer = List ('e', {pointer = Nil})})})})})})})}
    {pointer =
      List ('v',
       {pointer = List ('g', {pointer = List ('u', {pointer = Nil})})})}
```

Correct value 1 pt

```
  {pointer =
    List ('s',
     {pointer =
       List ('v',
        {pointer =
          List ('n',
           {pointer =
             List ('l',
              {pointer =
                List ('b',
                 {pointer =
                   List ('z',
                    {pointer =
                      List ('e',
                       {pointer =
                         List ('v',
                          {pointer =
                            List ('g', {pointer = List ('u', {pointer = Nil})})})})})})})})}
```

v Exercise 6: `filter` Completed, 10 pts

Found `filter` with compatible type.

Computing

```
  filter
    zero
    {pointer =
      List (-2,
       {pointer =
         List (-3,
          {pointer =
            List (4,
             {pointer =
               List (-4,
                {pointer =
                  List (4,
                   {pointer =
                     List (4,
                      {pointer =
                        List (-2,
                         {pointer =
                           List (3, {pointer = List (-5, {pointer = Nil})})})})})})})})})}
```

Correct value `{pointer = Nil}` 1 pt

Computing

```
  filter
    even
    {pointer =
      List (-2,
       {pointer =
         List (-5,
          {pointer =
            List (2,
             {pointer =
               List (3,
                {pointer =
                  List (-1,
                   {pointer =
                     List (0,
                      {pointer =
                        List (3,
                         {pointer =
                           List (2, {pointer = List (-1, {pointer = Nil})})})})})})})})})}
```

Correct value 1 pt

```
  {pointer =
    List (-2,
     {pointer =
       List (2, {pointer = List (0, {pointer = List (2, {pointer = Nil})})})})}
```

Computing

```
      List (3,
       {pointer =
         List (1,
          {pointer =
            List (4,
             {pointer =
               List (-5,
                {pointer =
                  List (-1,
                   {pointer =
                     List (-4,
                      {pointer =
                        List (4,
                         {pointer =
                           List (-3,
                            {pointer =
                              List (-3, {pointer = List (1, {pointer = Nil})})})})})})})})})})
Correct value                                                                          1 pt
  {pointer =
    List (3,
     {pointer =
       List (1,
        {pointer =
          List (4, {pointer = List (4, {pointer = List (1, {pointer = Nil})})})})})}
Computing
  filter
    odd
    {pointer =
      List (-2,
       {pointer =
         List (-4,
          {pointer =
            List (0,
             {pointer =
               List (3,
                {pointer =
                  List (-2,
                   {pointer =
                     List (3,
                      {pointer =
                        List (2,
                         {pointer =
                           List (-4, {pointer = List (0, {pointer = Nil})})})})})})})})})}
Correct value {pointer = List (3, {pointer = List (3, {pointer = Nil})})}             1 pt
Computing
  filter
    negative
    {pointer =
      List (-2,
       {pointer =
         List (1,
          {pointer =
            List (1,
             {pointer =
               List (0,
                {pointer =
                  List (2,
                   {pointer =
                     List (3,
                      {pointer =
                        List (1,
                         {pointer =
                           List (-4,
                            {pointer =
                              List (-3,
                               {pointer =
                                 List (-5,
                                  {pointer = List (3, {pointer = Nil})})})})})})})})})})})}
Correct value                                                                          1 pt
  {pointer =
    List (-2,
     {pointer =
       List (-4, {pointer = List (-3, {pointer = List (-5, {pointer = Nil})})})})}
Computing filter even {pointer = Nil}
Correct value {pointer = Nil}                                                          1 pt
Computing
  filter
    positive
    {pointer =
      List (-2,
       {pointer =
         List (1,
          {pointer =
            List (0,
             {pointer =
               List (1,
                {pointer = List (-2, {pointer = List (-5, {pointer = Nil})})})})})})}
Correct value {pointer = List (1, {pointer = List (1, {pointer = Nil})})}             1 pt
Computing filter negative {pointer = List (0, {pointer = Nil})}
```

```
    odd
    {pointer =
      List (-3,
       {pointer =
         List (0,
          {pointer =
            List (3,
             {pointer =
               List (-4,
                {pointer =
                  List (-3,
                   {pointer =
                     List (-4,
                      {pointer =
                        List (4, {pointer = List (-3, {pointer = Nil})})})})})})})})}
Correct value                                                                           1 pt
  {pointer =
    List (-3,
     {pointer =
       List (3, {pointer = List (-3, {pointer = List (-3, {pointer = Nil})})})})}
Computing
  filter
    zero
    {pointer =
      List (4,
       {pointer =
         List (3,
          {pointer =
            List (4,
             {pointer =
               List (1,
                {pointer = List (-3, {pointer = List (2, {pointer = Nil})})})})})})}
Correct value {pointer = Nil}                                                           1 pt
```