

```
Last login: Mon Mar  6 13:22:33 on ttys026
carbon:SamplePrograms$ cd Sec_01_1\:25pm/
carbon:Sec_01_1:25pm$ cat references.ml
(* Construct a circular structure of the form
```

```
c -->  1  -->  2  -->  3
      ^          |
      |          |
      +-----+
```

Write a function that returns the first n elements.

Each number above should be a pair with an int and a reference to the next pair.

*)

```
type box = Box of int * box ref
let dummy : box = Box (999, ref dummy)
let c =
  let ref_in_one = ref dummy in
  let one = Box (1, ref_in_one) in
  let three = Box (3, ref one) in
  let two = Box (2, ref three) in
  let () = ref_in_one := ref two
carbon:Sec_01_1:25pm$ utop
```

Welcome to utop version 1.14 (using OCaml version 4.01.0)!

Type #utop_help for help about using utop.

```
-( 18:00:00 )-< command 0 >-----{ counter: 0 }-
utop # #use "references.ml";;
File "references.ml", line 21, characters 0-0:
Error: Syntax error
-( 13:42:01 )-< command 1 >-----{ counter: 0 }-
utop # #use "references.ml";;
type box = Box of int * box ref
File "references.ml", line 14, characters 32-37:
Error: Unbound value dummy
-( 13:42:05 )-< command 2 >-----{ counter: 0 }-
utop # #use "references.ml";;
type box = Box of int * box ref
val dummy : box =
  Box (999,
    {contents =
      Box (999,
        {contents =
          Box (999,
            {contents =
              Box (999,
                {contents =
```

[illegible]

```
Error: This expression has type box ref but an expression was expected of type
      box
```

```
-( 13:42:41 )-< command 3 > { counter: 0 }-
utop # #use "references.ml";;
type box = Box of int * box ref
val dummy : box =
  Box (999,
    {contents =
      Box (999,
        {contents =
          Box (999,
            {contents =
```

[illegible]


```
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =
```

[illegible]

[illegible]

[illegible]

```
val c : box ref =
  {contents =
    Box (1,
      {contents =
        Box (2,
          {contents =
            Box (3,
              {contents =
                Box (1,
```

```
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =
```



```
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =  
Box (2,  
{contents =  
Box (3,  
{contents =  
Box (1,  
{contents =
```

[illegible]

```
utop # #use "references.ml";;
```

```
val dummy : box =
```

[illegible]

[illegible]

[illegible]

[illegible]

```
val c' : box =  
    Box (1,  
        {contents =  
            Box (2,  
                {contents =  
                    Box (3,  
                        {contents =  
                            Box (1,  
                                {contents =  
                                    Box (2,  
                                        {contents =  
                                            Box (3,
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]