

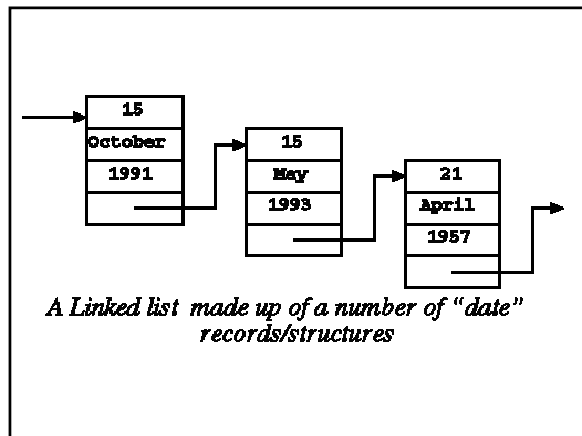
COMP205 IMPERATIVE LANGUAGES

8. COMPOUND (HIGHER LEVEL) DATA TYPES 4 --- LINKED LISTS AND UNIONS

- 1) Linked records/structures
- 2) Linked lists, examples in Ada and C
- 3) Processing linked lists

LINKED RECORDS/STRUCTURES

- One of the most useful applications of records is to define organisations of data in which distinct items are linked together using *pointers* or *reference values*.
- The simplest general form of linked records is a *linked list* (more complex forms include trees of various types).
- In a linked list each record includes a pointer to a following record.
- The final pointer is a *null pointer* (defined using the reserved word NULL in both Ada and C) indicating that the last record in the linked list has been reached.



```
procedure ADA_LINKED_LIST_EX is
  type DATE_T; -- Incomplete declaration
  type DATEPTR_T is access DATE_T;
  type DATE_T is record
    DAY : integer; MONTH : string(1..9);
  end record;
  procedure CREATE_BIRTHDAY_RECORD (DAY: in integer;
    MONTH: in string; YEAR: in integer;
    NEWPTR: in out DATEPTR_T) is
  begin
    NEWPTR := new DATE_T'(DAY,MONTH,YEAR,NEWPTR);
  end CREATE_BIRTHDAY_RECORD;
  CREATE_BIRTHDAY_RECORD(15, "October",
    1991, BIRTHDAYPTR);
  CREATE_BIRTHDAY_RECORD(15, "May",
    1993, BIRTHDAYPTR);
  CREATE_BIRTHDAY_RECORD(21, "April",
    1957, BIRTHDAYPTR);
end ADA_LINKED_LIST_EX;
```

C LINKED LIST EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>

typedef struct date {
  int day;
  char month[9],
  int year;
  struct date *next;
} DATE_T, *DATE_PTR_T;

DATE_PTR_T createBdayStruct(int, int, int);

void main(void){
----- rest of code
```

```
void main(void) {
  DATE_PTR_T birthdayPtr=NULL, newPtr=NULL;

  birthdayPtr=createBdayStruct(15, "October",
    1991);
  newPtr=createBdayStruct(15, "May", 1993);
  newPtr->next=birthdayPtr;
  birthdayPtr=newPtr;
  newPtr=createBdayStruct(21, "April", 1957);
  newPtr->next=birthdayPtr;
  birthdayPtr=newPtr;
}
```

```
DATE_PTR_T createBdayStruct(int day, int month,
                           int year) {
    DATE_PTR_T newPtr;

    if ((newPtr = (DATE_PTR_T)
         (malloc(sizeof(DATE_T)))!=NULL) {
        printf("Insufficient space\n");
        exit(-1);
    }

    newPtr->day = day;
    strcpy(newPtr->month,month);
    newPtr->year = year;
    newPtr->next = NULL;

    return(newPtr);
}
```

SUMMARY

- 1) Linked records/structures
- 2) Linked lists, examples in Ada and C
- 3) Processing linked lists