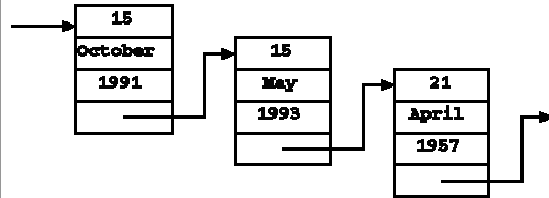


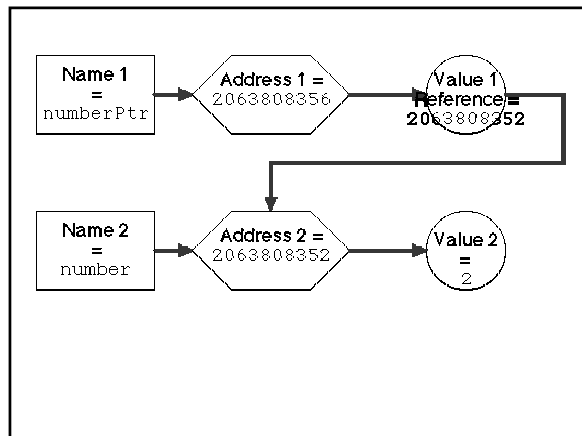
COMP205 IMPERATIVE LANGUAGES

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



A Linked list made up of a number of "date" records/structures



```

#include <stdio.h>

void main(void)
{
  int n, *nptr;

  n = 2;
  nptr = &n;
  printf("n = %d\n",n);
  printf("address of n = %d\n",nptr);
  printf("nptr = %d\n",nptr);
  printf("address of nptr = %d\n",&nptr);
  printf("value pointed at = %d\n", *nptr);

  n = 4;
  nptr = &n;
  printf("n = %d\n",n);
  printf("address of n = %d\n",&n);
  printf("nptr = %d\n",nptr);
  printf("address of nptr = %d\n",&nptr);
  printf("value pointed at = %d\n", *nptr);
}
  
```

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, char*, 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,
                           char *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);
}

```

PROCESSING LINKED LISTS

- The standard mechanism for processing linked lists is to step through the list.
- Ada Example:

```

with CS_IO ; use CS_IO ;

procedure ADA_LINKED_LIST_EX is
--- Data declarations ----
--- Create birthday record code ----

    procedure OUTPUT (LINKPTR: in DATEPTR_T) is
begin
    put("Birthday = ");
    put(LINKPTR.DAY, width => 2);
    put(" ");put(LINKPTR.MONTH(1..9));
    put(LINKPTR.YEAR, width => 3);new_line;
end OUTPUT; _EX ;

```

```

procedure OUTPUT_BIRTHDAY_RECORDS (
    START_PTR: in DATEPTR_T) is
    LINKPTR : DATEPTR_T:= STARTPTR;
begin
    OUTPUT(LINKPTR); LINKPTR := LINKPTR.NEXT;
    OUTPUT(LINKPTR); LINKPTR := LINKPTR.NEXT;
    OUTPUT(LINKPTR);
end OUTPUT_BIRTHDAY_RECORDS;

begin
    --- Call create birthday record code ----
    OUTPUT_BIRTHDAY_RECORDS(BIRTHDAYPTR);
end ADA_LINKED_LIST_EX ;

```

```

Birthday = 21 April      57
Birthday = 15 May       93
Birthday = 15 October   91

```

C EXAMPLE

```

void outputBirthdayStructures (DATE_PTR_T birthdayPtr) {
DATE_PTR_T linkPtr = birthdayPtr;

output(linkPtr);
linkPtr = linkPtr->next;
output(linkPtr);
linkPtr = linkPtr->next;
output(linkPtr);
}

void output(DATE_PTR_T linkPtr) {
printf("Birthday = %d %s %d\n",linkPtr->day,
    linkPtr->month,linkPtr->year);
}

```

SUMMARY

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