addRecord

Homework #6 …
Ehsan Kourkchi

addRecord (pointer to record start, array uname, array uaddr, int uyob, array utelno)

Define a ***<pointer to record>*** called ***<current>***
copy ***<start>*** into the ***<current>***

if (***<current>*** is not ***NULL***):

    while (the ***<next field of the record whose address is in current>*** is not ***NULL***):
        copy ***<next field of the record whose address is in current>*** into the ***<current>***

    Allocate space for a new record and store its address in the ***<next field of the record whose address is in current>***
    copy ***<next field of the record whose address is in current>*** into the ***<current>***

else:
    Allocate space for a new record and store its address in the ***<start>***
    copy ***<start>*** into the ***<current>***

Copy ***<NULL>*** into the ***<next field of the record whose address is in current>***

Copy the ***<uname array>*** into the ***<name field of the record whose address is in current>***

Copy the ***<uaddr array>*** into the ***<addresss field of the record whose address is in current>***

Copy the ***<uyob int>*** into the ***<yearofbirth field of the record whose address is in current>***

Copy the ***<utelno array>*** into the ***<telno field of the record whose address is in current>***

```
int addRecord (struct record *start,char uname[],char uaddr[],int uyob, char utelno[]) {

  struct record *current;
  current = start;
  int i;

  if (current != NULL)
  {

    while (current->next != NULL)
    {
      current = current->next;
    }

  current->next = (struct record *)malloc(sizeof(struct record));
  current = current->next;


  }
  else
  {

    start = (struct record *)malloc(sizeof(struct record));
    current = start;

  }

current->next = NULL;

for (i=0; i<25; i++)  current->name[i] = uname[i];

for (i=0; i<80; i++)  current->address[i] = uaddr[i];

current->yearofbirth = uyob;

for (i=0; i<15; i++)  current->telno[i] = utelno[i];

  }
```

deleteRecord

deleteRecord (pointer to record start, array uname)

Define a ***<pointer to record>*** called ***<current>***

Define a ***<pointer to record>*** called ***<temp>***

copy ***<start>*** into the ***<current>***

while (***<current>*** is not ***NULL***):

if (***<name field of the record whose address is in current>*** is ***<uname array>):***

if (<current> is <start>):
copy ***<next field of the record whose address is in current>*** into the ***<start>***
Delete the record whose address is in <current>
copy <start> into the <current>

else:

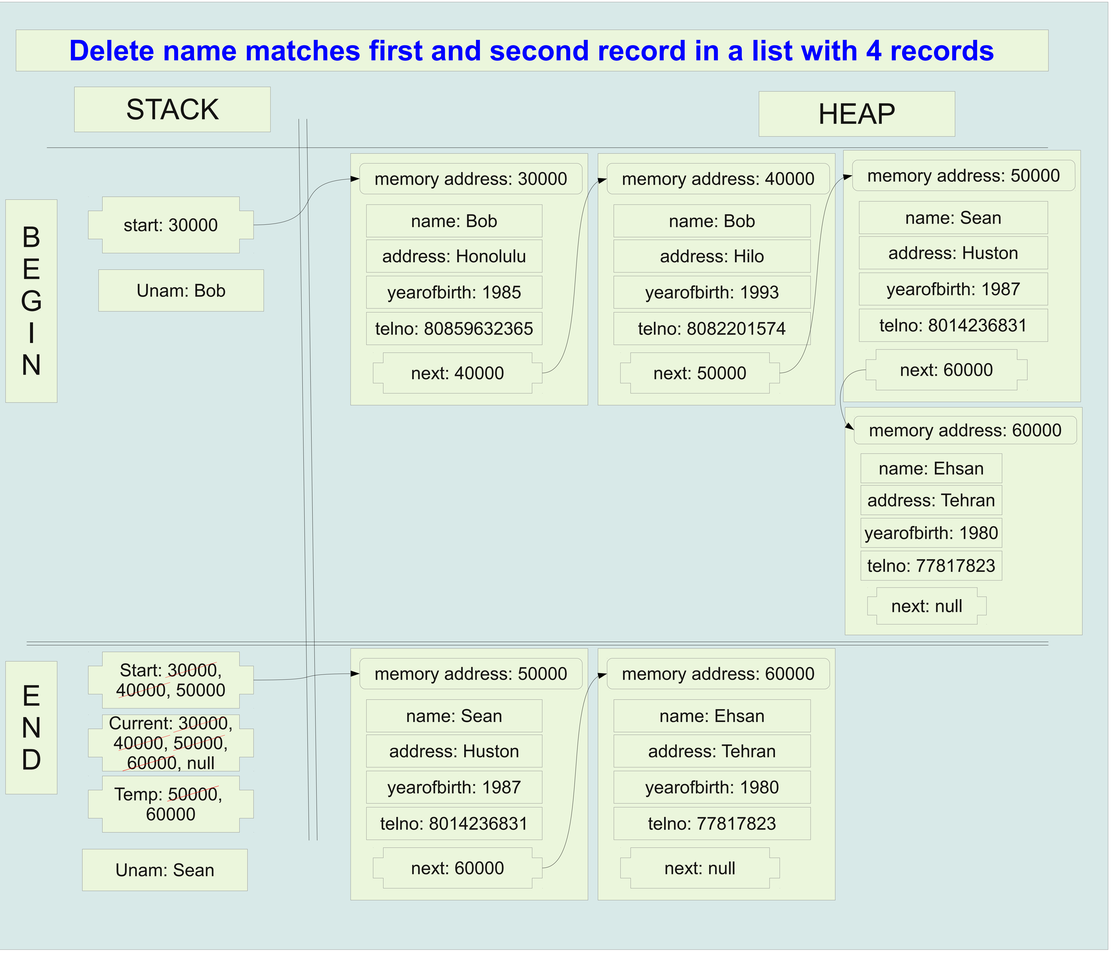
copy ***<next field of the record whose address is in current>*** into the ***<next field of the record whose address is in temp>***
Delete the record whose address is in ***<current>***
copy <next field of the record whose address is in temp> into the <current>

else*:*
copy <current> into the <temp>
copy ***<next field of the record whose address is in current>*** into the ***<current>***

Homework #6 …
Ehsan Kourkchi

```
int deleteRecord(struct record *start,char uname[]) {

  struct record *current;
  struct record *temp;
  current = start;
  int i;

  while(current != NULL) {

    if (strcmp(current->name,uname) == 0)
    {

      if (current = start)
      {
     start = current-> next;
     free(current);
     current = start;
      }
      else
      {

     temp->next = current->next;
     free(current);
     current = temp->next;

      }

    }
    else
    {
      temp = current;
      current = current->next;
    }

  }

}
```


Add record to list which has two records
STACK
HEAP
BaGIN
start: 30000
Unam: Ehsan
Uaddr: Tehran
Uyob: 1980
Utelno: 77817823
memory address: 30000
name: Bob
address: Honolulu
yearofbirth: 1985
telno: 80859632365
next: 40000
memory address: 40000
name: Lauren
address: Hilo
yearofbirth: 1993
telno: 8082201574
next: null
END
start: 30000
Current: 30000 40000, 50000
Unam: Ehsan
Uaddr: Tehran
Uyob: 1980
Utelno: 77817823
memory address: 30000
name: Bob
address: Honolulu
yearofbirth: 1985
telno: 80859632365
next: 40000
memory address: 40000
name: Lauren
address: Hilo
yearofbirth: 1993
telno: 8082201574
next: 50000
memory address: 50000
name: Ehsan
address: Tehran
yearofbirth: 1980
telno: 77817823
next: null


Delete name matches first and second record in a list with 4 records
STACK
HEAP
BEGIN
start: 30000
Unam: Bob
memory address: 30000
name: Bob
address: Honolulu
yearofbirth: 1985
telno: 80859632365
next: 40000
memory address: 40000
name: Bob
address: Hilo
yearofbirth: 1993
telno: 8082201574
next: 50000
memory address: 50000
name: Sean
address: Huston
yearofbirth: 1987
telno: 8014236831
next: 60000
memory address: 60000
name: Ehsan
address: Tehran
yearofbirth: 1980
telno: 77817823
next: null
END
Start: 30000, 40000, 50000
Current: 30000, 40000, 50000, 60000, null
Temp: 50000, 60000
Unam: Sean
memory address: 50000
name: Sean
address: Huston
yearofbirth: 1987
telno: 8014236831
next: 60000
memory address: 60000
name: Ehsan
address: Tehran
yearofbirth: 1980
telno: 77817823
next: null