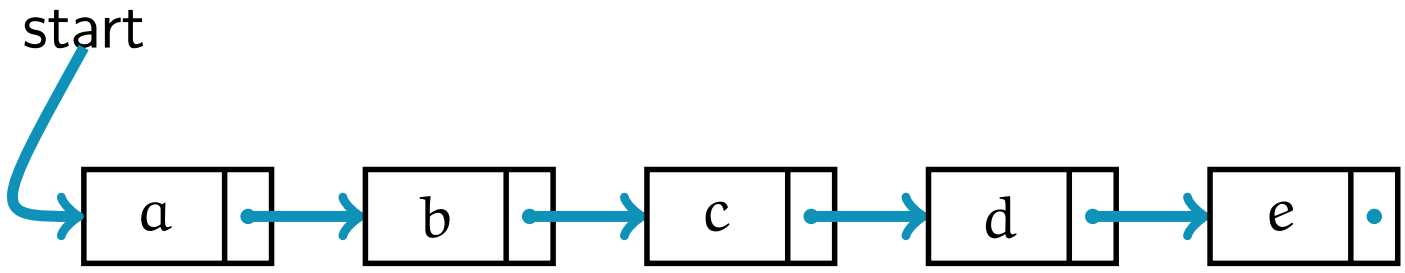
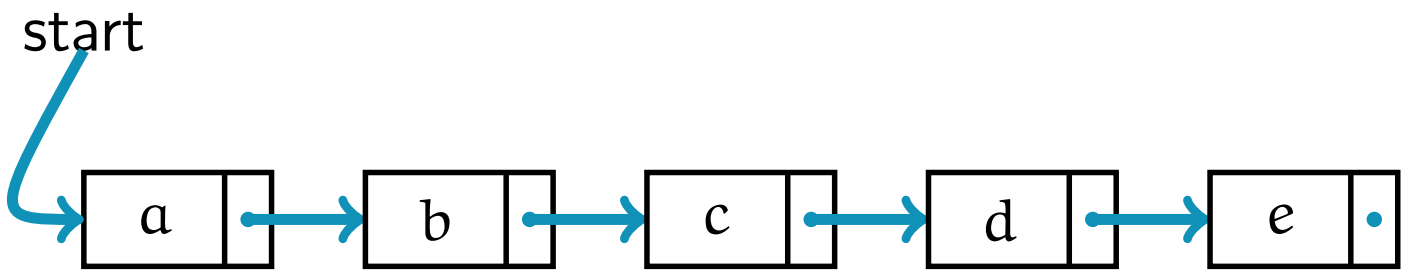
Linked Lists
- Search

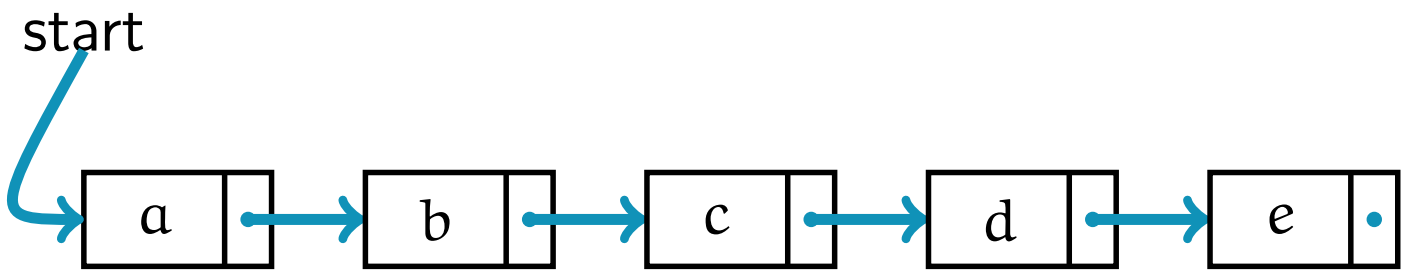


Linked Lists - Search



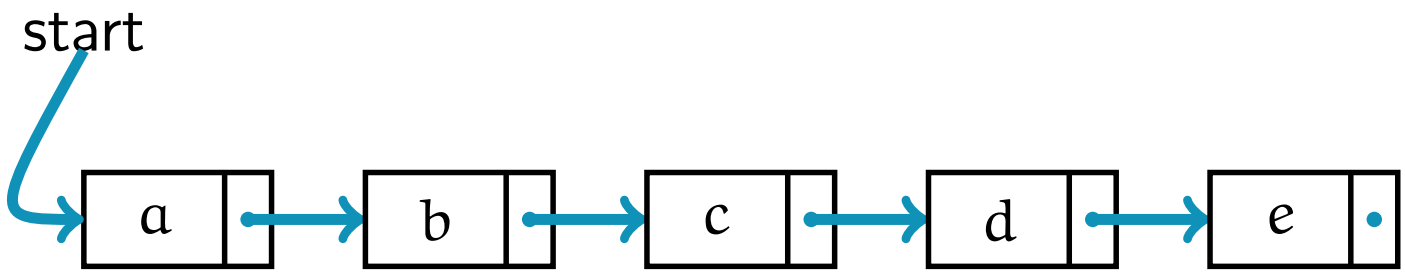

```
search(struct node *str, int element){




}
```

Linked Lists - Search



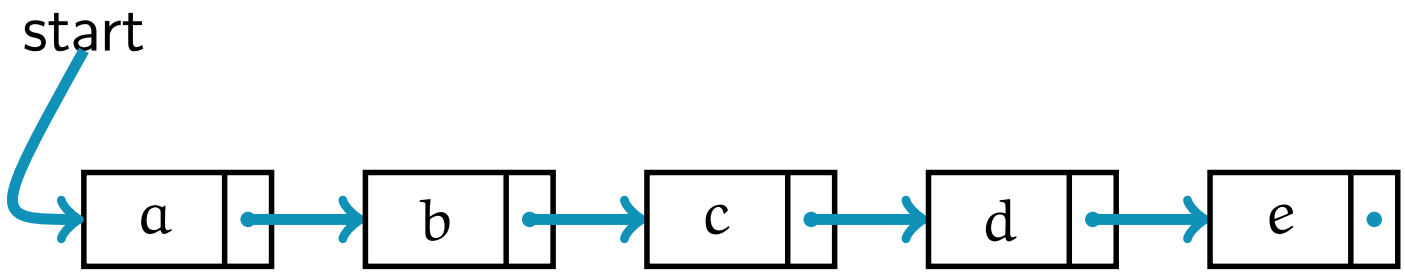

```
search(struct node *str, int element){
      struct node *temp;
      temp = str;

}
```

Linked Lists - Search



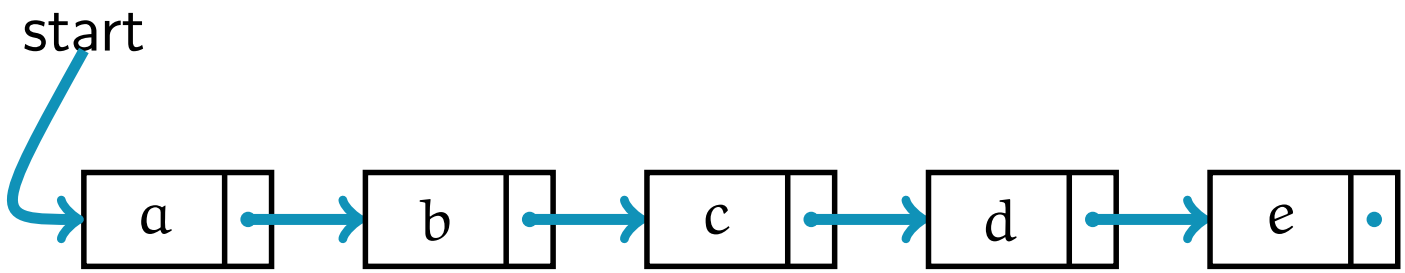

```
search(struct node *str, int element){
      struct node *temp;
      temp = str;
      while(temp != NULL){


      }



}
```

Linked Lists - Search



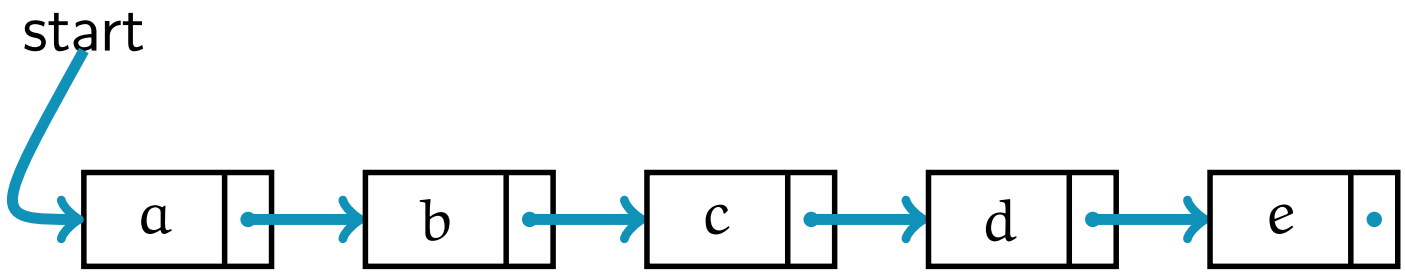

```
search(struct node *str, int element){
      struct node *temp;
      temp = str;
      while(temp != NULL){
            if(temp->data == element)

      }



}
```

Linked Lists - Search



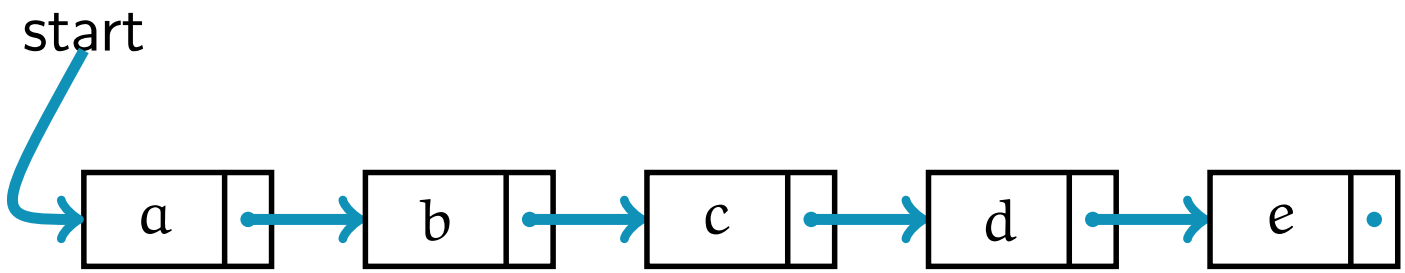

```
search(struct node *str, int element){
      struct node *temp;
      temp = str;
      while(temp != NULL){
            if(temp->data == element)  break;

      }



}
```

Linked Lists - Search



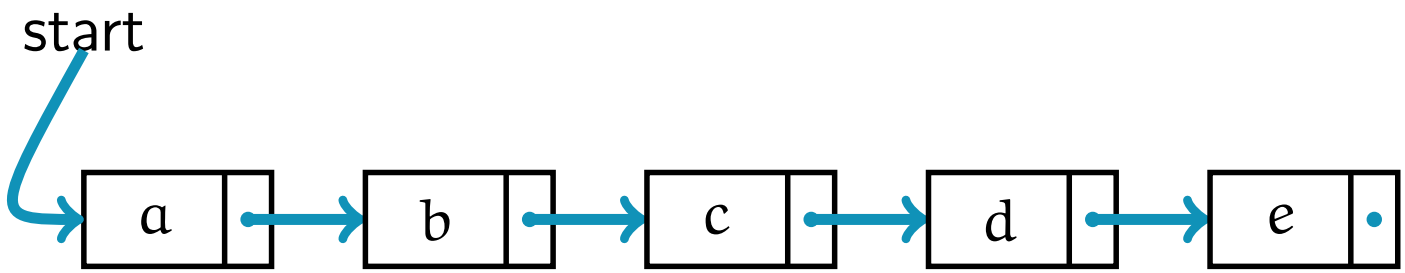

```
search(struct node *str, int element){
      struct node *temp;
      temp = str;
      while(temp != NULL){
            if(temp->data == element)  break;
            else temp = temp->ptr;
      }



}
```

Linked Lists - Search



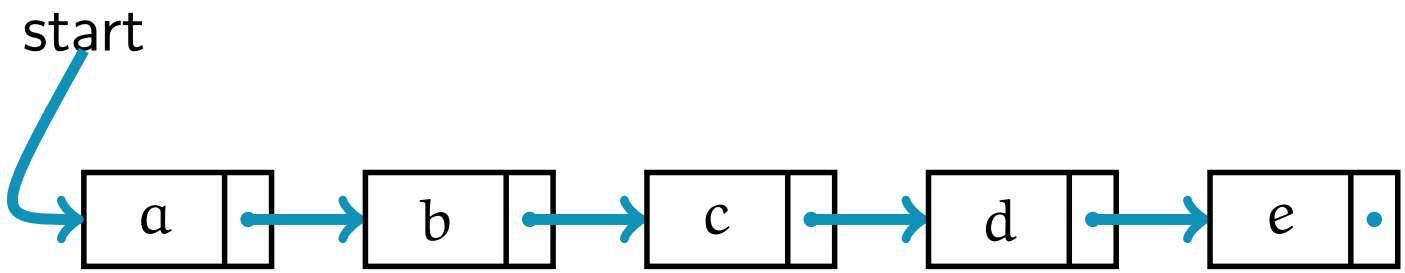

```
search(struct node *str, int element){
      struct node *temp;
      temp = str;
      while(temp != NULL){
            if(temp->data == element)  break;
            else temp = temp->ptr;
      }
      if(temp == NULL)



}
```

Linked Lists - Search




```
search(struct node *str, int element){
      struct node *temp;
      temp = str;
      while(temp != NULL){
            if(temp->data == element)  break;
            else temp = temp->ptr;
      }
      if(temp == NULL)
            print(‘‘Not Found’’);


}
```

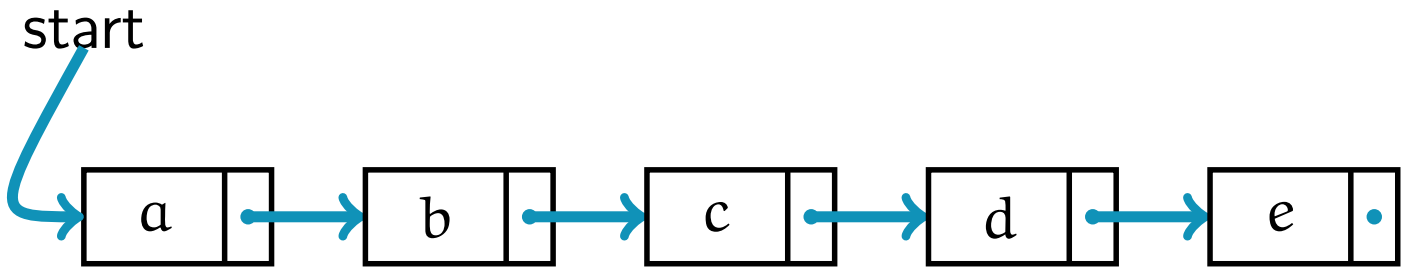
Linked Lists - Search



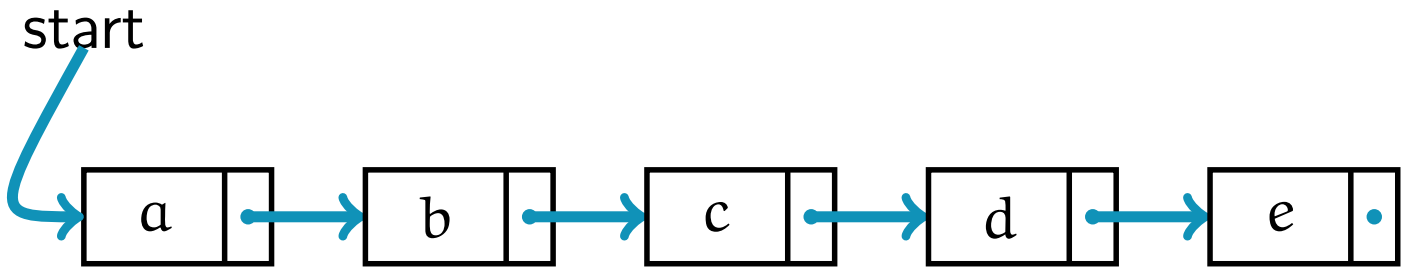

```
search(struct node *str, int element){
      struct node *temp;
      temp = str;
      while(temp != NULL){
            if(temp->data == element)  break;
            else temp = temp->ptr;
      }
      if(temp == NULL)
            print(‘‘Not Found’’);
      else
            printf(‘‘Found’’);
}
```

Linked Lists

- Add at Start

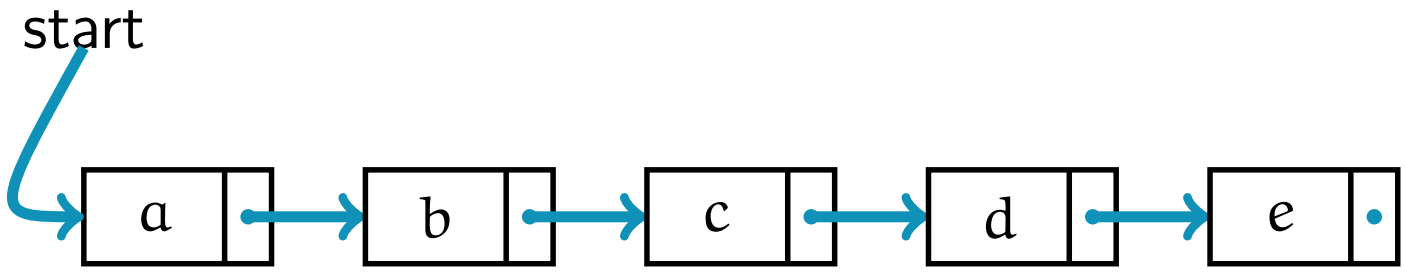


Linked Lists
- Add at Start



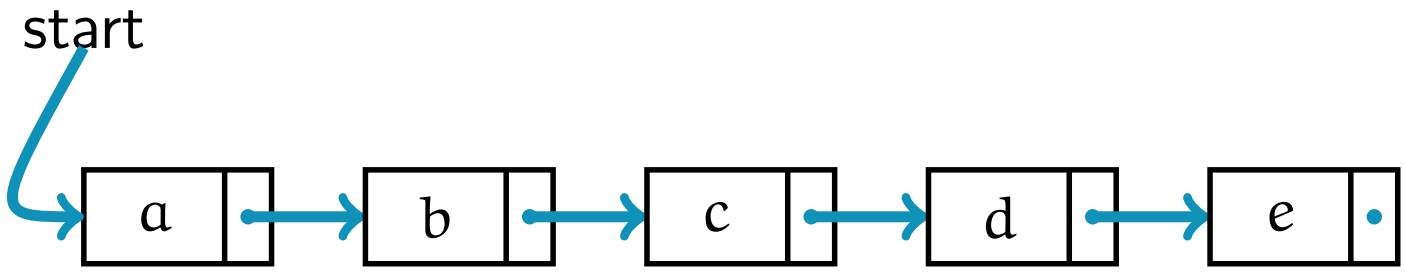

```
addStart(struct node **str, int data) {




}
```

Linked Lists
- Add at Start



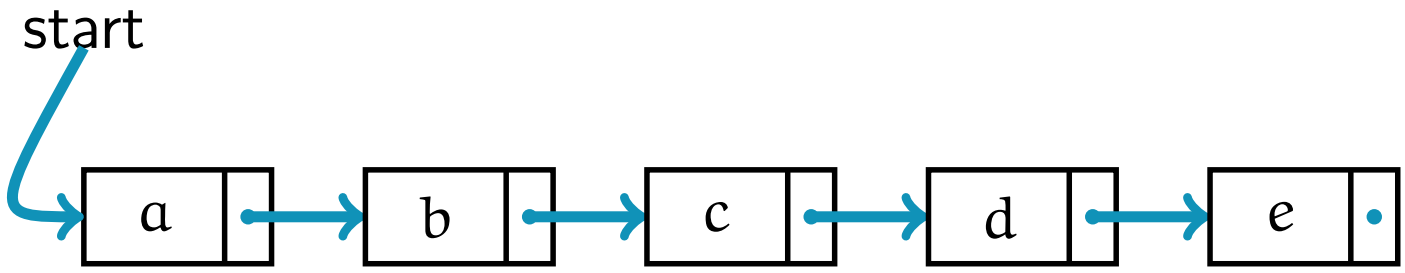

```
addStart(struct node **str, int data) {
      struct node *newNode;
      newNode = createNode();




}
```

Linked Lists
- Add at Start



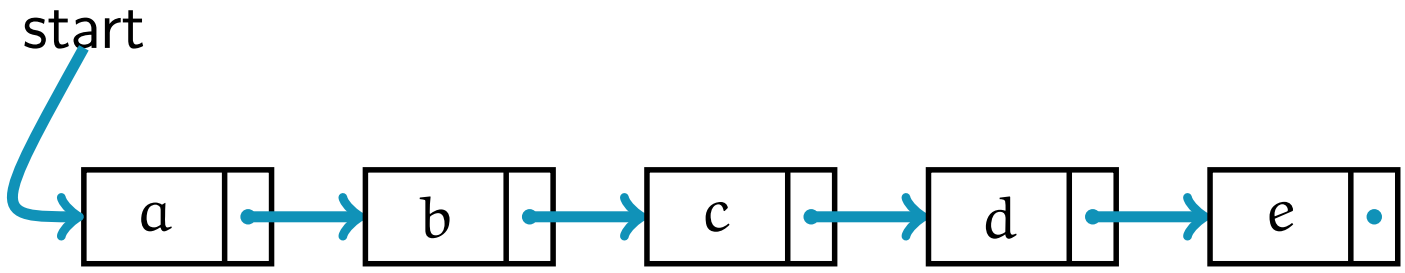

```
addStart(struct node **str, int data) {
      struct node *newNode;
      newNode = createNode();
      if(newNode == NULL)
      {
            printf("Cannot create new node."); return;
      }



}
```

Linked Lists
- Add at Start



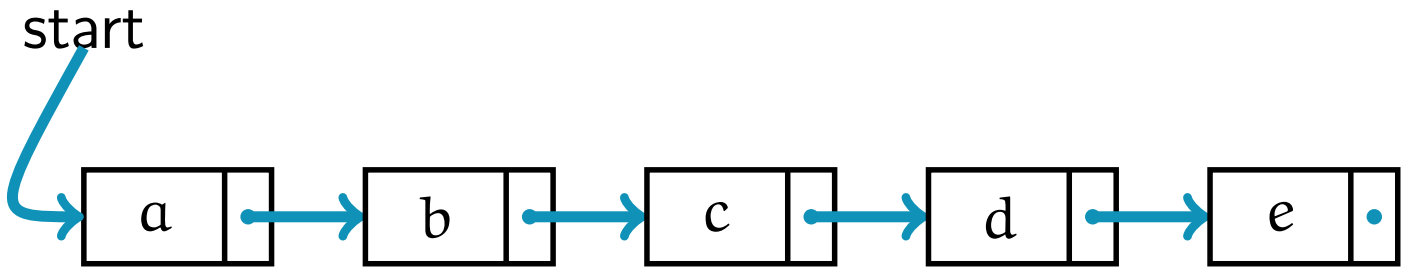

```
addStart(struct node **str, int data) {
      struct node *newNode;
      newNode = createNode();
      if(newNode == NULL)
      {
            printf("Cannot create new node."); return;
      }
      newNode->data = data;


}
```

Linked Lists
- Add at Start



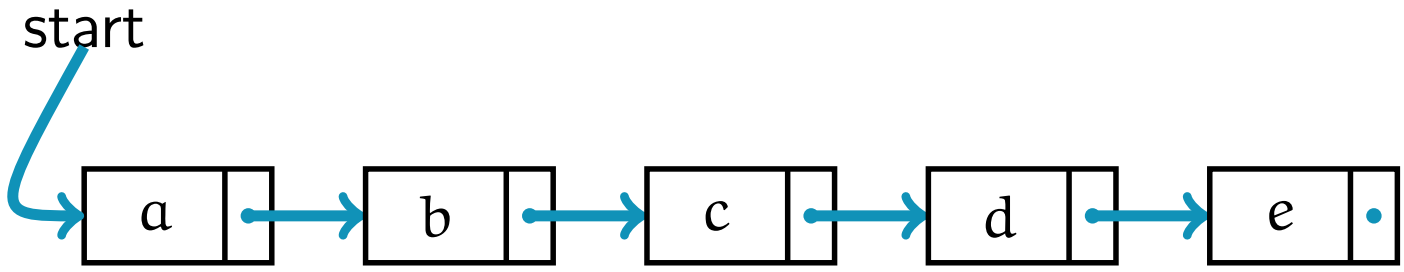

```
addStart(struct node **str, int data) {
      struct node *newNode;
      newNode = createNode();
      if(newNode == NULL)
      {
            printf("Cannot create new node."); return;
      }
      newNode->data = data;
      newNode->ptr = *str;

}
```

Linked Lists
- Add at Start

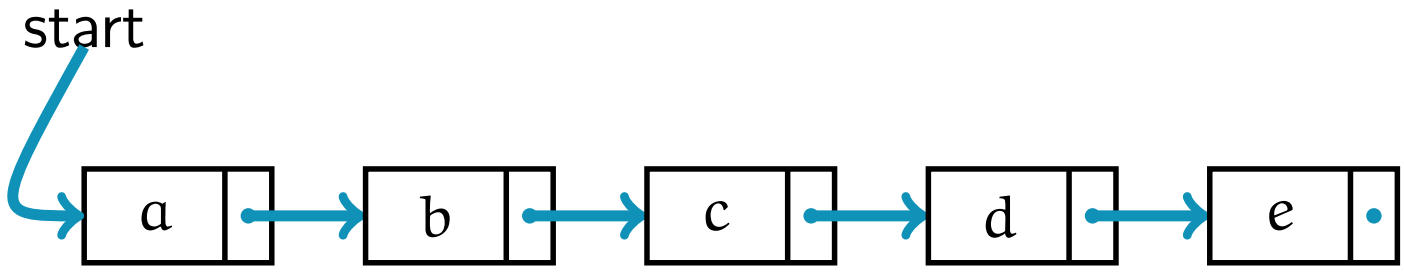



```
addStart(struct node **str, int data) {
      struct node *newNode;
      newNode = createNode();
      if(newNode == NULL)
      {
            printf("Cannot create new node."); return;
      }
      newNode->data = data;
      newNode->ptr = *str;
      *str = newNode;
}
```

Linked Lists – Delete at Start

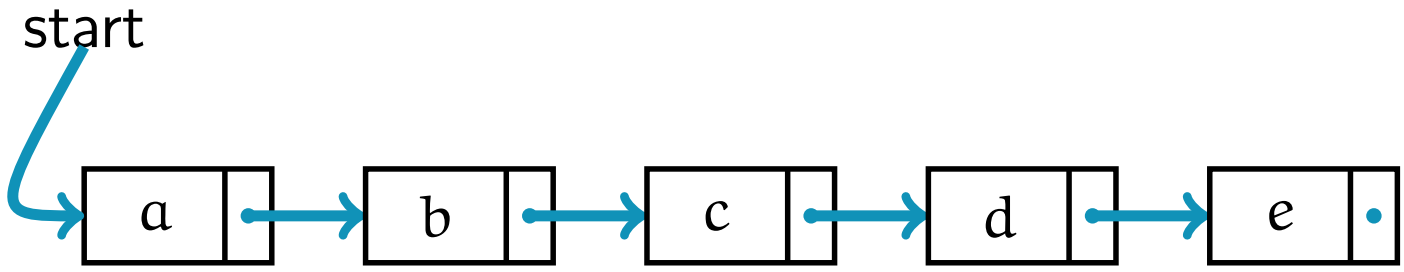


Linked Lists – Delete at Start



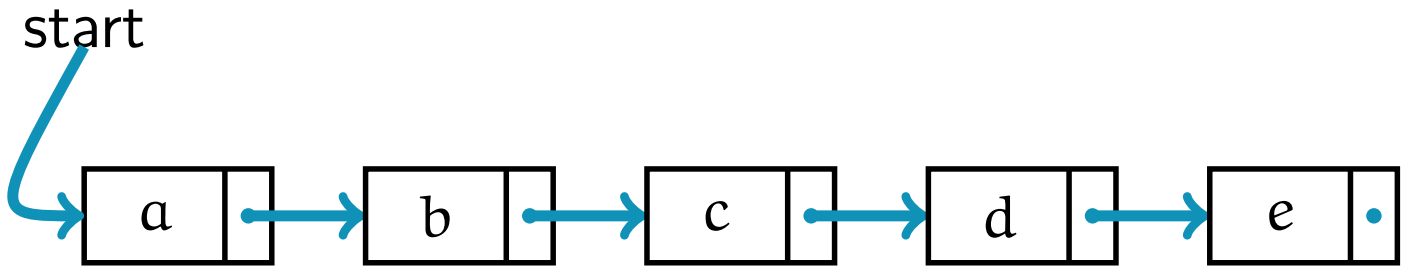

```
deleteStart(                        ) {




}
```

Linked Lists – Delete at Start



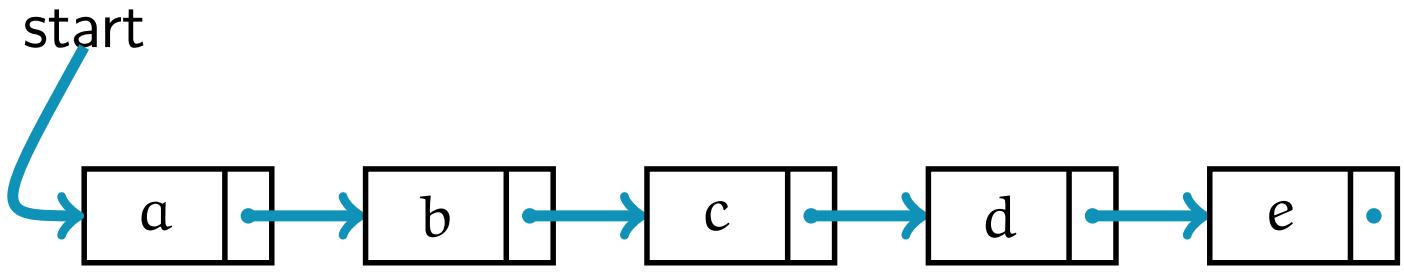

```
 /* deleteStart(&start) */
deleteStart(                        ) {




}
```

Linked Lists – Delete at Start



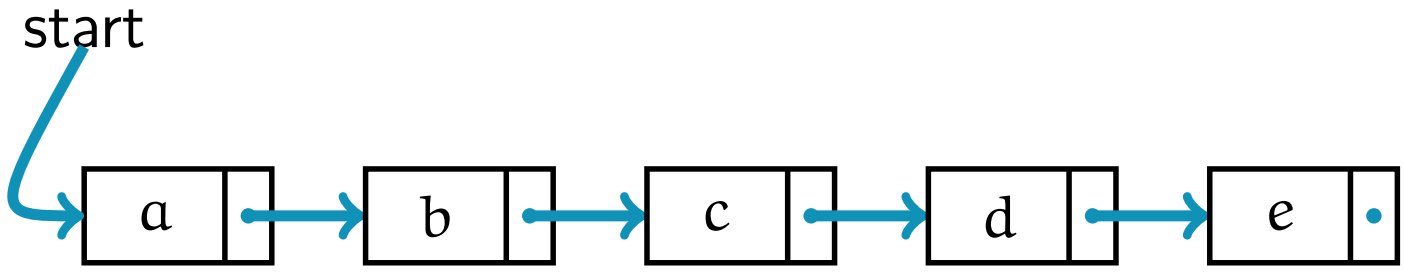

```
 /* deleteStart(&start) */
deleteStart(struct node **str) {




}
```

Linked Lists – Delete at Start



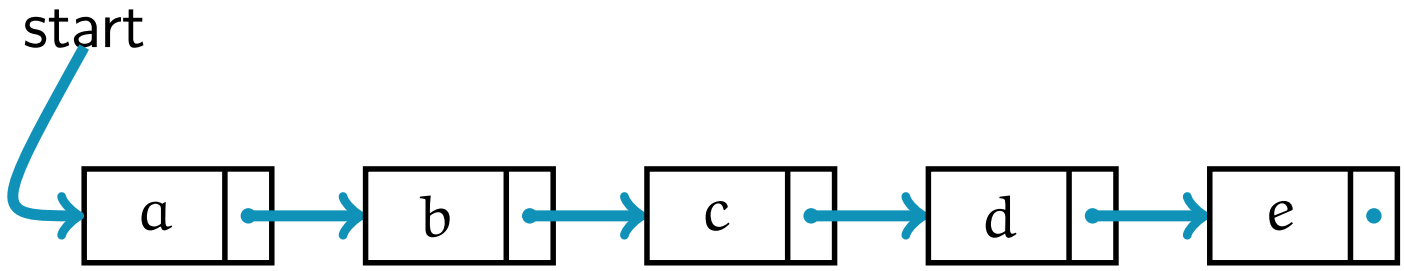

```
 /* deleteStart(&start) */
deleteStart(struct node **str) {
                        /* empty link list */




}
```

Linked Lists – Delete at Start



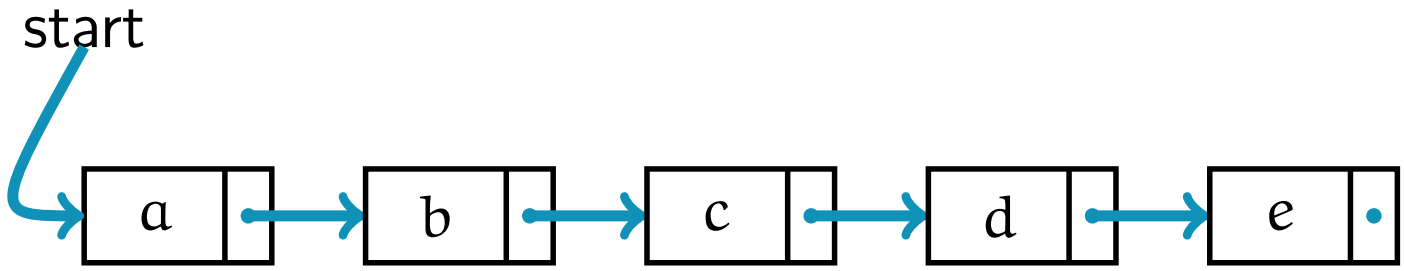

```
 /* deleteStart(&start) */
deleteStart(struct node **str) {
  if(*str == NULL) { /* empty link list */

  }




}
```

Linked Lists – Delete at Start



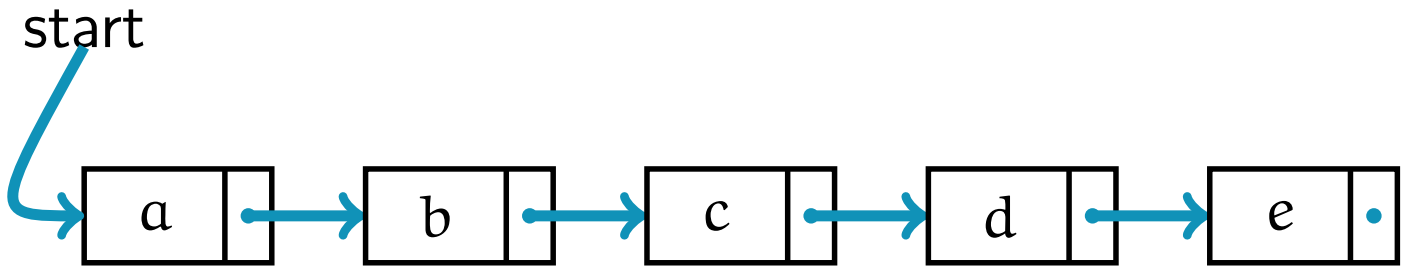

```
 /* deleteStart(&start) */
deleteStart(struct node **str) {
  if(*str == NULL) { /* empty link list */
    return;
  }




}
```

Linked Lists – Delete at Start



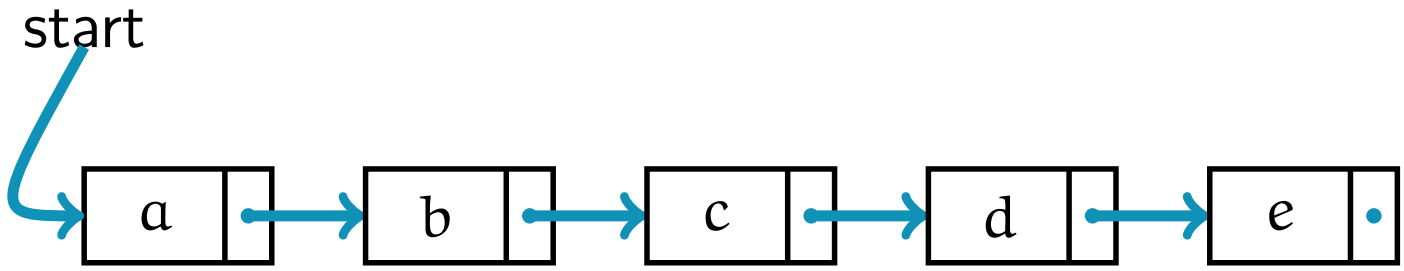

```
 /* deleteStart(&start) */
deleteStart(struct node **str) {
  if(*str == NULL) { /* empty link list */
    return;
  }
  else {



  }
}
```

Linked Lists – Delete at Start



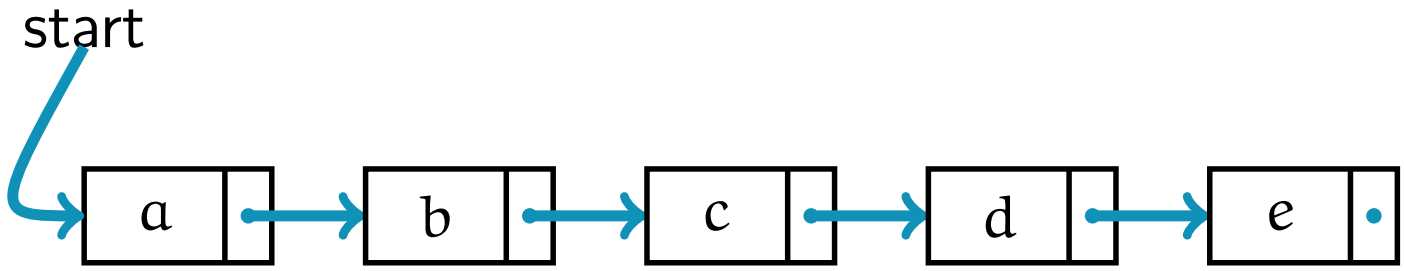

```
 /* deleteStart(&start) */
deleteStart(struct node **str) {
  if(*str == NULL) { /* empty link list */
    return;
  }
  else {
    temp = *str;



  }
}
```

Linked Lists – Delete at Start



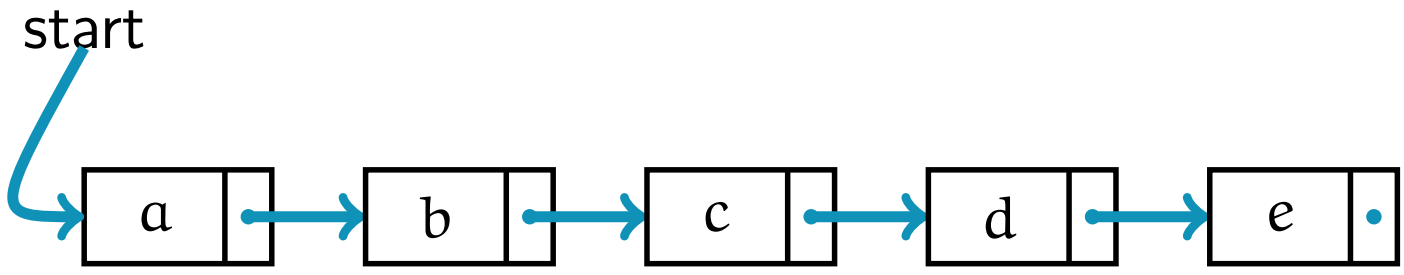

```
 /* deleteStart(&start) */
deleteStart(struct node **str) {
  if(*str == NULL) { /* empty link list */
    return;
  }
  else {
    temp = *str;
    *str = (*str)->ptr;

  }
}
```

Linked Lists – Delete at Start

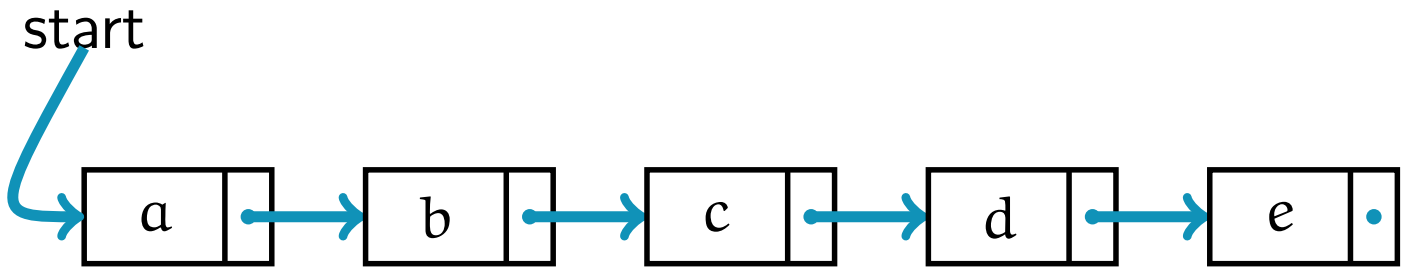



```
 /* deleteStart(&start) */
deleteStart(struct node **str) {
  if(*str == NULL) { /* empty link list */
    return;
  }
  else {
    temp = *str;
    *str = (*str)->ptr;
    free(temp);
  }
}
```

Linked Lists – Delete at End

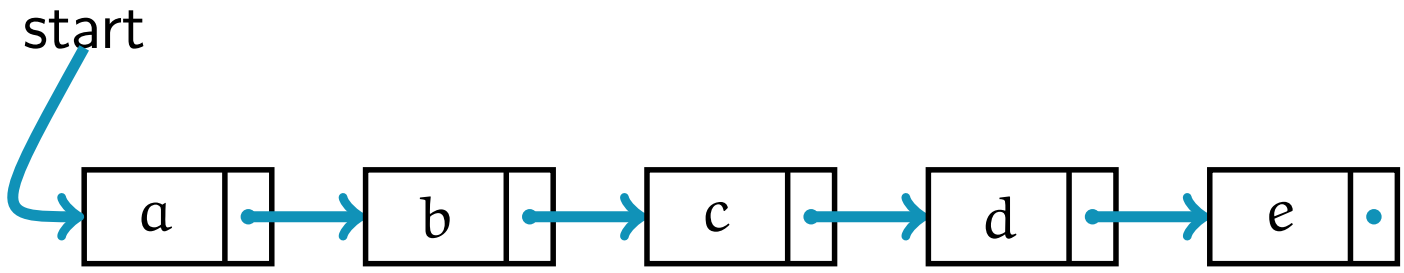


Linked Lists – Delete at End



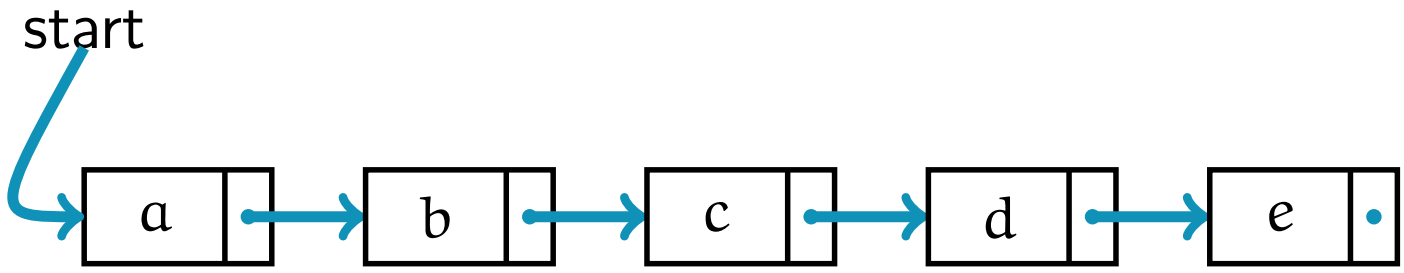

```
deleteEnd(                          ) {




}
```

Linked Lists – Delete at End



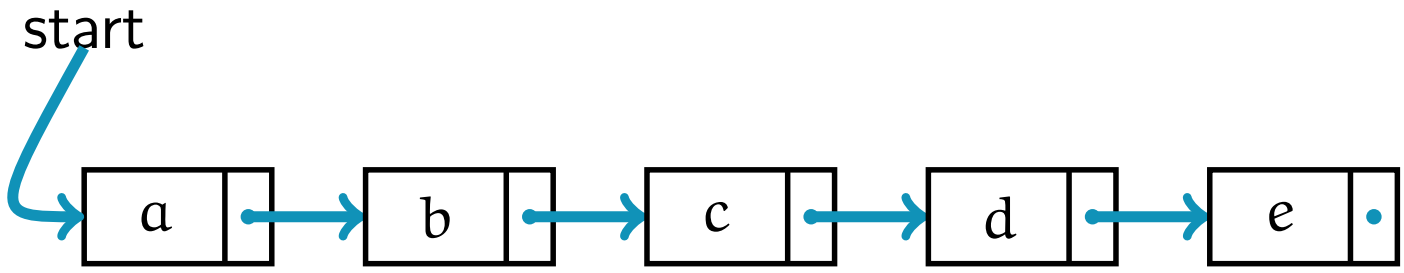

```
 /* start or &start */
deleteEnd(                    ) {




}
```

Linked Lists – Delete at End



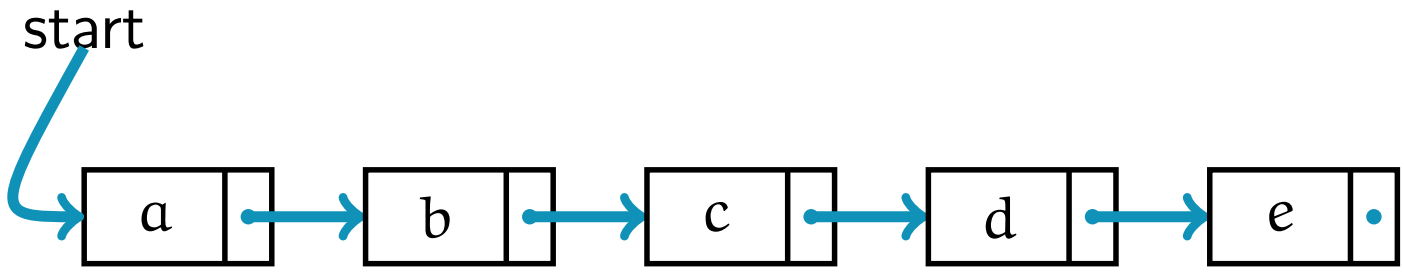

```
 /* start or &start -- list with one node?  */
deleteEnd(                          ) {




}
```

Linked Lists – Delete at End



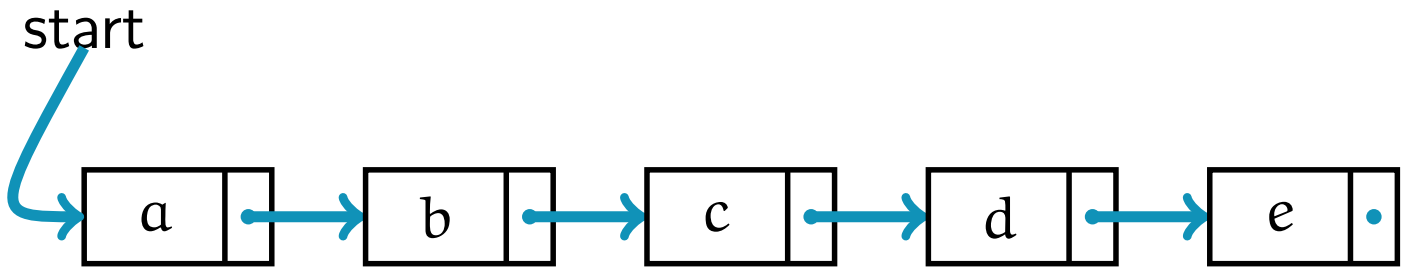

```
deleteEnd(struct node **str) {




}
```

Linked Lists – Delete at End



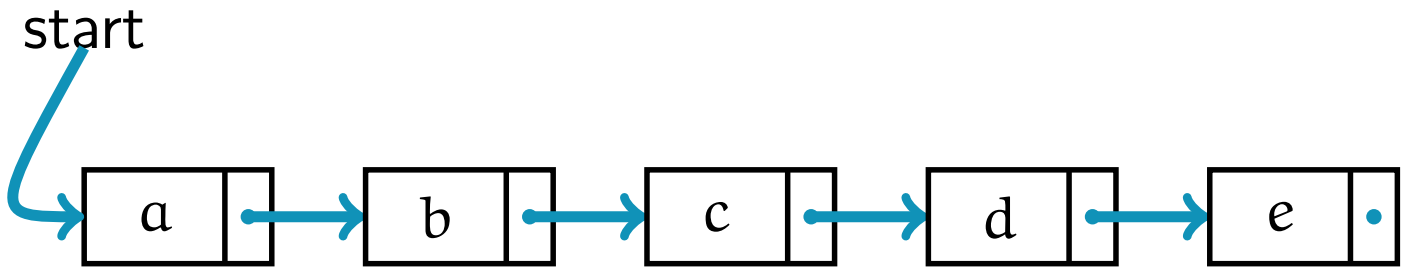

```
 /* find second last node */
deleteEnd(struct node **str) {




}
```

Linked Lists – Delete at End



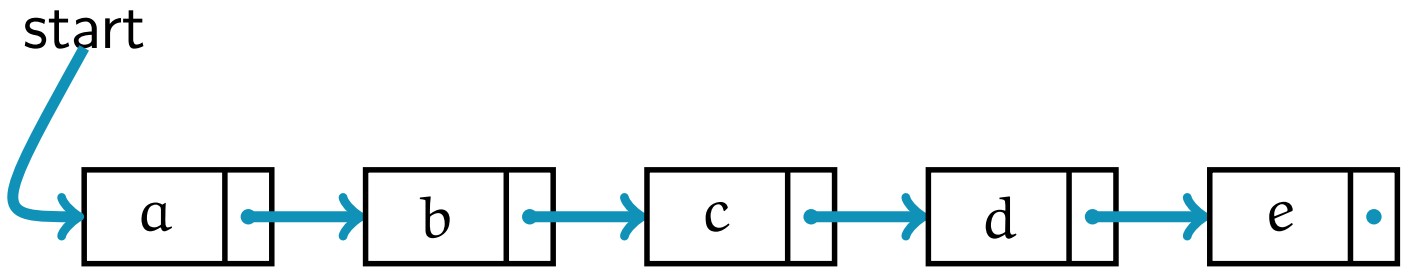

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;




}
```

Linked Lists – Delete at End



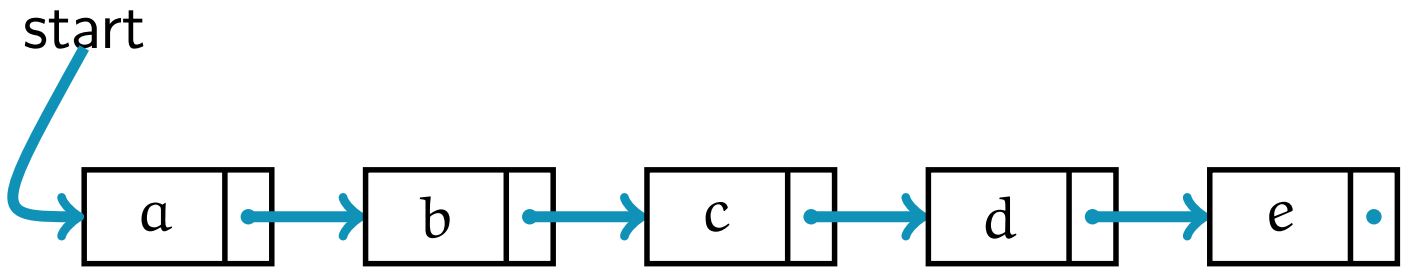

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str; /* the first node */




}
```

Linked Lists – Delete at End



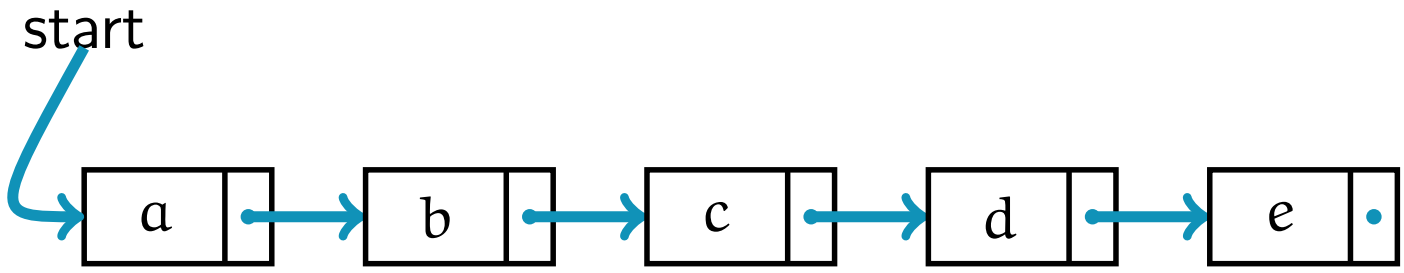

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL; /* node before first */




}
```

Linked Lists – Delete at End



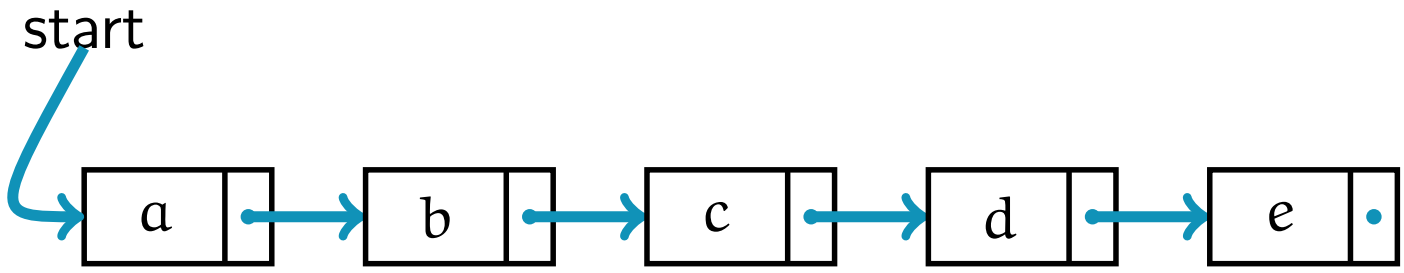

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
                              /* empty list */




}
```

Linked Lists – Delete at End



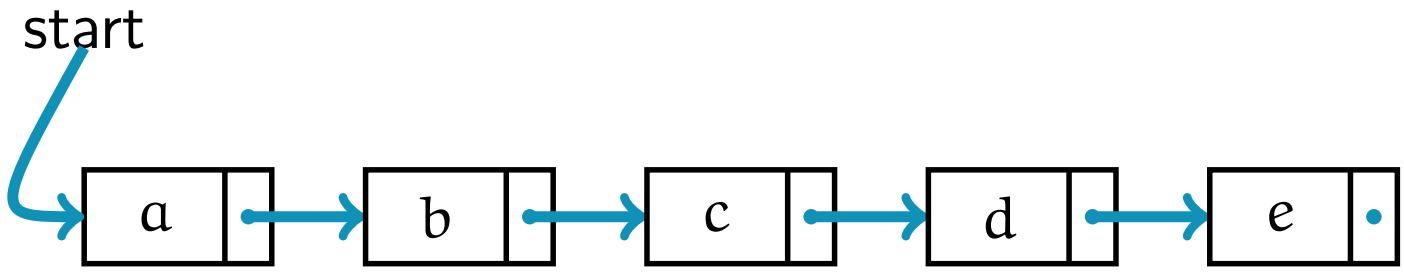

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */




}
```

Linked Lists – Delete at End



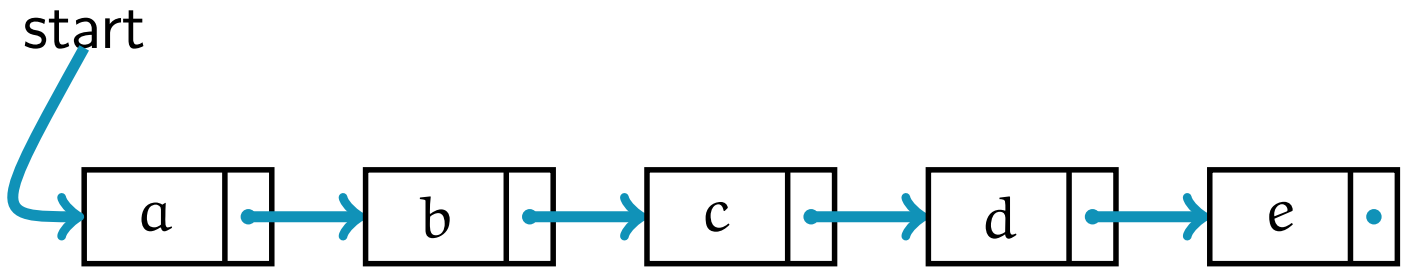

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
                                /* find last node */



}
```

Linked Lists – Delete at End



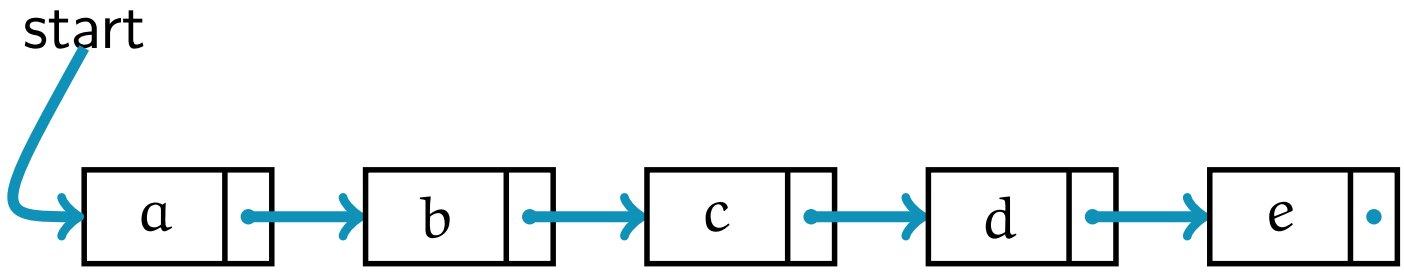

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */

  }

}
```

Linked Lists – Delete at End



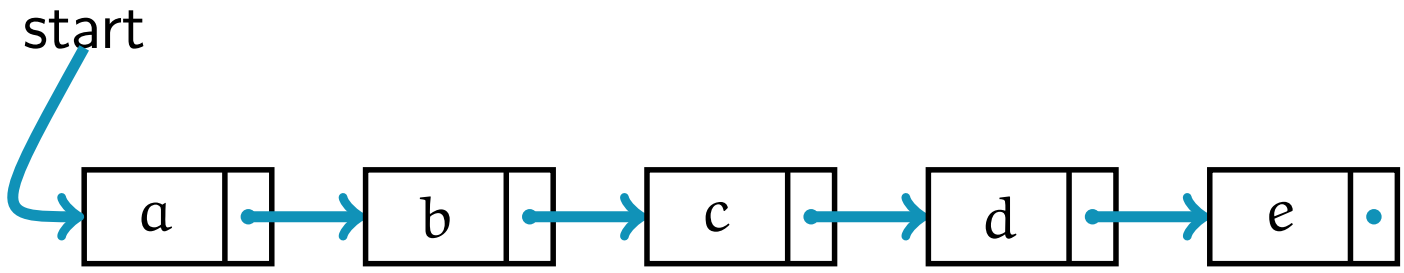

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }

}
```

Linked Lists – Delete at End



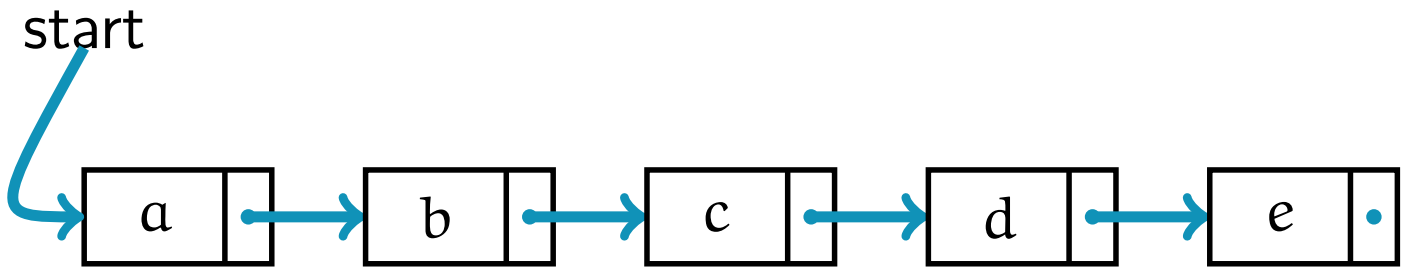

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
                /* free last node */
}
```

Linked Lists – Delete at End



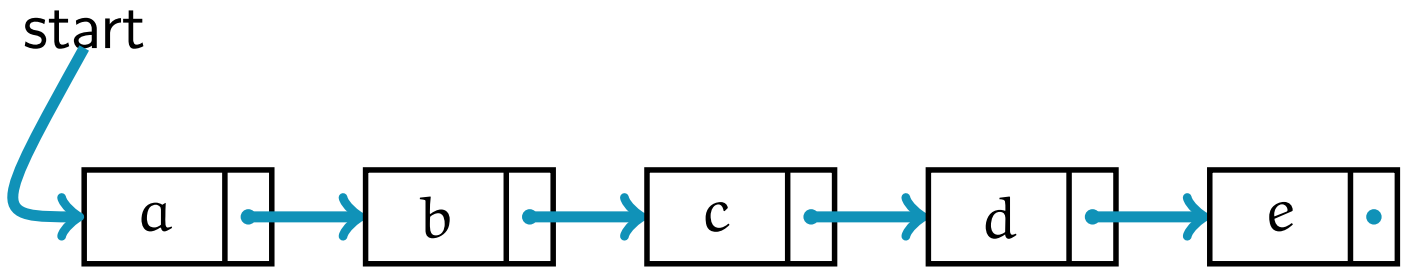

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
}
```

Linked Lists – Delete at End



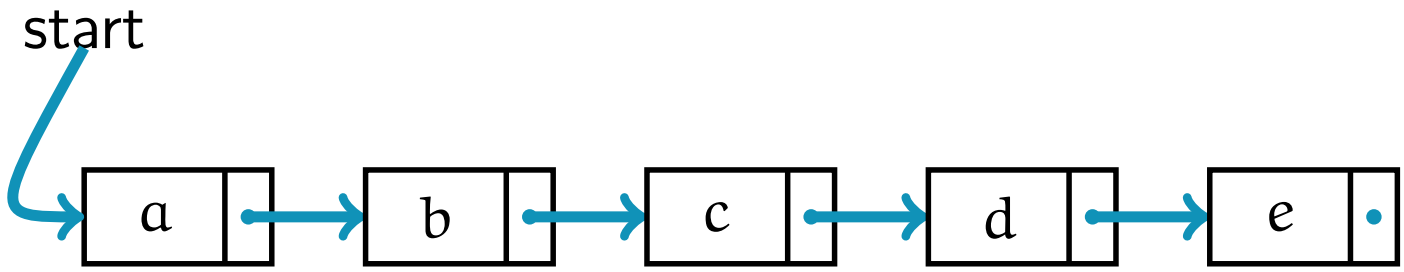

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
  /* what more?  */
}
```

Linked Lists – Delete at End



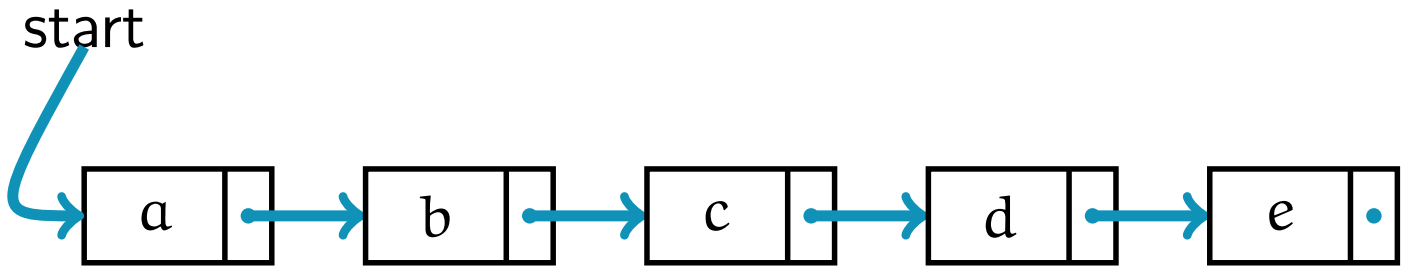

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
  prev->ptr = NULL;
}
```

Linked Lists – Delete at End



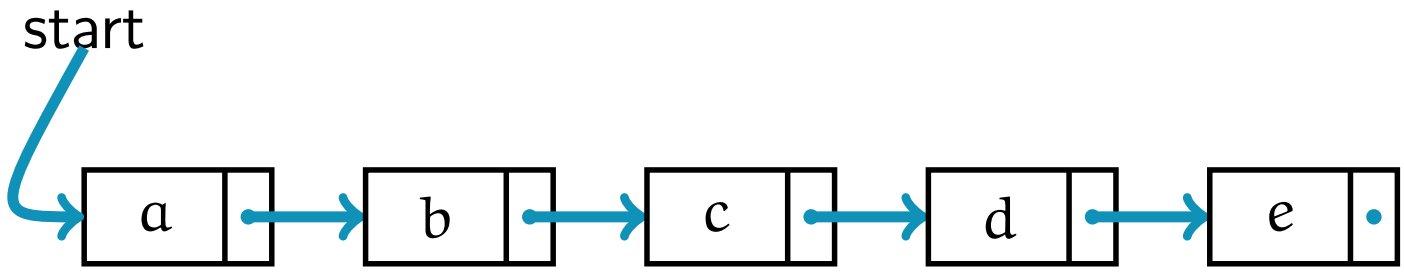

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
  prev->ptr = NULL; /* can 'prev' be NULL? */
}
```

Linked Lists – Delete at End



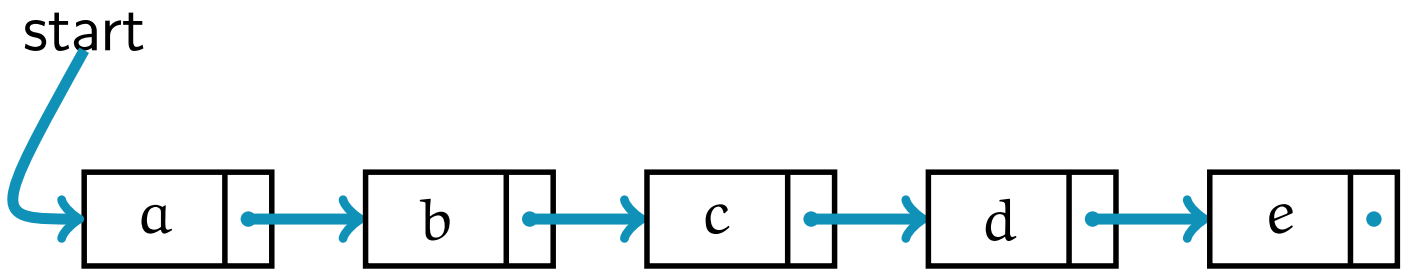

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
  prev->ptr = NULL; /* list with one node?  */
}
```

Linked Lists – Delete at End



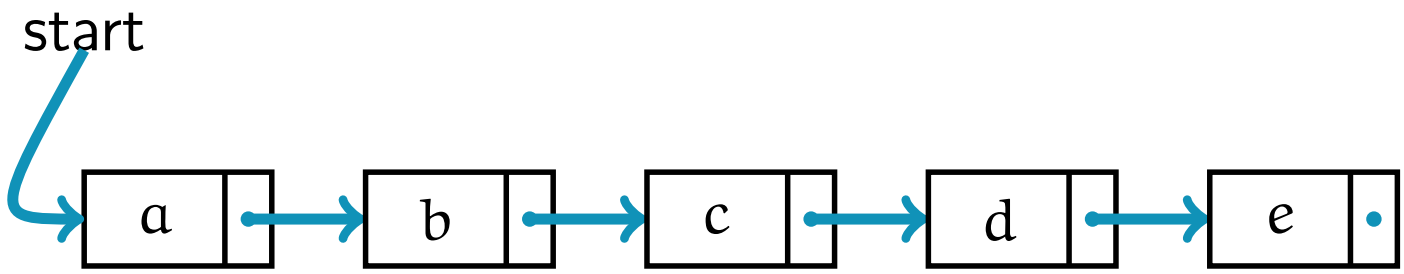

```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
  if(prev != NULL) prev->ptr = NULL;
}
```

Linked Lists – Delete at End




```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
  if(prev != NULL) prev->ptr = NULL;
  else
}
```

Linked Lists – Delete at End




```
 /* find second last node */
deleteEnd(struct node **str) {
  struct node *temp, *prev;
  temp = *str, prev = NULL;
  if(temp == NULL) return; /* empty list */
  while(temp->ptr != NULL) { /* find last node */
    prev = temp; temp = temp->ptr;
  }
  free(temp); /* free last node */
  if(prev != NULL) prev->ptr = NULL;
  else *str = NULL;
}
```