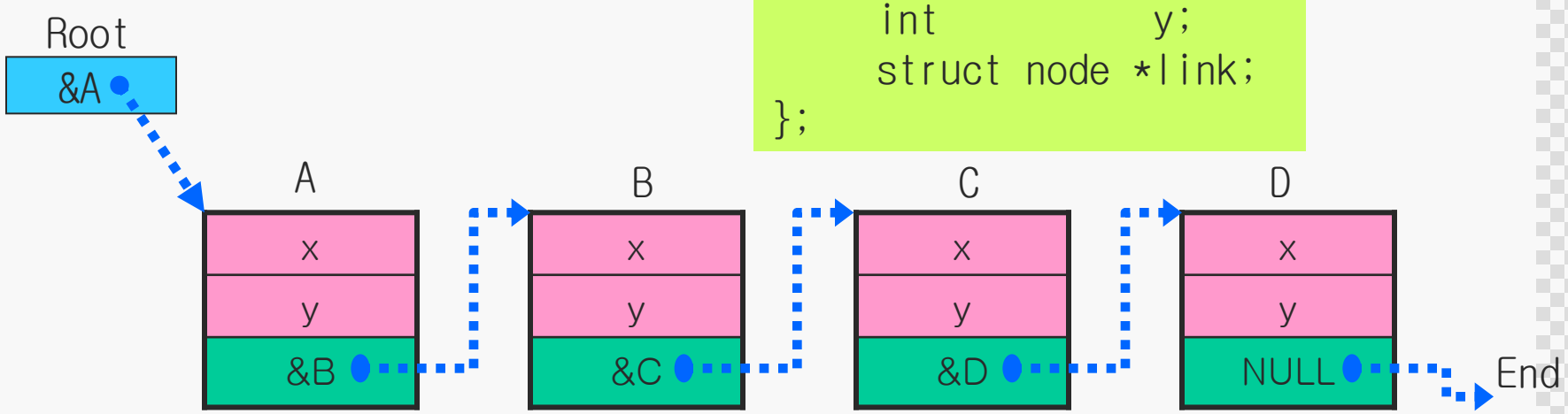
Project #9(Pjt09_linked_list)

Linked List: concept

```
struct  node {
      int           x;
      int           y;
      struct node *link;
};
```




```
struct  node  *Root;
struct  node   A, B, C, D;

Root = &A;
A.link = &B;
B.link = &C;
C.link = &D;
D.link = NULL;
```

```
struct  node  *cp;

for (cp = Root; cp != NULL; cp = cp->link)
   printf("x=%d, y=%d₩n", cp->x, cp->y);


for (cp = Root; cp != NULL; cp = (*cp).link)
   printf("x=%d, y=%d₩n", (*cp).x, (*cp).y);
```

Project #9(Pjt09_linked_list)

- Data Structure for Linked List

```
struct node {
     int            data;
     struct node   *link;
};

extern struct node *Head;

void   insert_node_head(struct node *np),
       insert_node_tail(struct node *np),
       insert_node_ascn(struct node *np),
       free_node(struct node *np), tour_list(), free_list();
struct node *get_node();
```

Project #9(Pjt09_linked_list)

Data Structure for Linked List

```
#include <stdlib.h>
struct node *Head = NULL;

struct node *get_node()
{
    struct node *cp;

    cp = (struct node *)malloc(sizeof(struct node));
    return(cp);
}

void free_node(struct node *np)
{
    free(np);
}
```

Project #9(Pjt09_linked_list)

- Data Structure for Linked List

```
void insert_node_head(struct node *np)
{
    if (!Head) {
        Head = np;
        np->link = NULL;
    }
    else {
        np->link = Head;
        Head = np;
    }
}
```

Project #9(Pjt09_linked_list)

- Data Structure for Linked List

```
void insert_node_tail(struct node *np)
{
    struct node *cp;

    if (!Head)
        Head = np;
    else {
        for (cp = Head; cp->link != NULL; cp = cp->link)
            ;
        cp->link = np;
    }
    np->link = NULL;
}
```

Project #9(Pjt09_linked_list)

- Data Structure for Linked List

```
void insert_node_ascn(struct node *np)
{
    struct node  *cp, *pp;

    if (!Head) {
        Head = np;
        np->link = NULL;
    }
    else {
        for (cp = Head, pp = NULL; cp != NULL && cp->data < np->data;
                                                    pp = cp, cp = cp->link)
            ;
        if (pp == NULL) {
            np->link = Head;
            Head = np;
        } else {
            np->link = pp->link;
            pp->link = np;
        }
    }
}
```

Project #9(Pjt09_linked_list)

- Data Structure for Linked List

```
void tour_list()
{
    struct node *cp;

    printf("₩n")
    for (cp = Head; cp != NULL; cp = cp->link)
         printf("-->%c ", cp->data);
    printf("₩n")
}

void free_list()
{
    struct node *cp;

    for (  ; Head != NULL; ) {
        cp = Head;
        Head = cp->link;
        free_node(cp);
    }
}
```

Project #9(Pjt09_linked_list)

Sample Task to Test Linked List

```
void  app_list(char *ap)
{
    char          buf[8], how =  'a';
    struct node *np;

    if (ap) how = *ap;    // how = ap[0]
    while(1) {
       printf("> ");
       if (fgets(buf, 8, stdin) == NULL)
           break;
       np = get_node();
       np->data = buf[0];     // use only the first character in buf
        switch(how) {
              case 'h' :insert_node_head(np); break;
              case 't' :insert_node_tail(np); break;
              default :insert_node_ascn(np); //  'a'
         }
    }
    tour_list();
    free_list();
}
```

Project #9(Pjt09_linked_list)

```
int is_prime(int  n)
{
  int  i;
  for (i = 2; i <= n/2; i++)
     if ((n % i) == 0)
        return(0);
  return(1);
}
void app_prime(char  *ap)
{
  int  t = 2000, count = 0, n;

  if (ap) t = atoi(ap);
  for (n = 2; n <= t; n++) {
     if  (is_prime(n)) {
       count++;
       printf("%d  is a prime number !!!₩n", n);
     }
  }
  printf("count=%d₩n", count);
}
```

Project #9(Pjt09_linked_list)

```
#include <stdio.h>
#include <string.h>
main()
{
    char  cmd[128], *cp, *ap;
    int   n = 0;

    uart_init();
    sei();
    while(1) {
       printf( “$ ” );
       if (fgets(cmd, sizeof(cmd), stdin) == NULL)
           break;
       if ((cp = strtok(cmd,  “₩n₩r₩t  “)) == NULL) continue
       ap = strtok(NULL,  “₩n₩r₩t  “);
       if      (!strcmp(cmd,  “prime” )) app_prime(ap);
       else if (!strcmp(cmd,  “list” ))  app_list(ap);
       else                              printf( “Unknown command...₩n” );
    }
    printf( “logout, good bye !!!₩n” );
    while(1);
 }
```