Lecture 9
C Programming Language

Comments on final

- Closed book, closed notes, closed everything
- All readings are important
- There will be an emphasis on issues discussed in class
- Format of questions:
  - define / describe a certain constructs in C
    (what is a macro - in your own words)
  - determine whether a code is legal
    (will it compile)
  - determine whether a given code will cause a run-time error
  - determine output of function
  - given some code, describe what it does
- References to functions you implemented in your homework

Pointers to struct

- How many errors (if any) are in this function?

```
Void InsertFront(List *ilist, int val)
{
        Listitem * newitem;
        newitem.next = ilist.head;
        newitem.data = val;
        ilist.head = newitem;
}
```

Pointers to struct

- Are these versions ok ?

```
void InsertFront(List *ilist, int val)
{
   Listitem * newitem;

   newitem = (Listitem *) malloc(sizeof(Listitem));
   (*newitem).next = (*ilist).head;
   (*newitem).data = val;
   (*ilist).head = newitem;
}

void InsertFront(List *ilist, int val)
{
   Listitem * newitem;
   newitem = (Listitem *) malloc(sizeof(Listitem));
   newitem->next = ilist->head;
   newitem->data = val;
   ilist->head = newitem;
}
```

Pointers to struct

- What is wrong here ?

```
void Init(int **arr, int sz_n,int sz_m)
{
        int j,k;
        arr = (int **) malloc(sz_n*sizeof(int *));
        for (j=0;j<sz_n;j++)
        {
          arr[j]= (int *)malloc(sz_m*sizeof(int));
          for (k=0;k<sz_m;k++)
                arr[j][k]=0;
        }
}
main()
{
  int ** a;
  Init(a,10,5);
  printf(“ entry[3][3] is %d\n”,a[3][3]);
}
```

More Errors...

- Errors ?

```
float divide(int numer, int denom)
{
        if (denom == 0)
        {
           fprintf(stderr,”divide by 0\n”);
           exit(1);
        }
        return (numer / denom);
}
```

- Compile Errors ? Run time Errors ?

```
void strcat(char *s, char *t)
{
        while (*s)
            s++;
        while (*s++ = *t++);
}
```

What does this function do ?

- What does this function do ?
- Any compile-time errors / run-time errors?

```
void mystery(char * s)
{
        char * t;
        int c;
        for (t=s+(strlen(s)-1);s<t;s++, t--)
        {
           c = *s;
           *s = *t;
           *t = c;
        }
}
…
mystery(s);
printf(‘%s\n”,s);
```

Review Questions

- What is the difference between internal and external (global) variables ?
- What is the difference between static and automatic variables ?
- If a variable is declared outside a function block with a modifier **static**, does it have internal linkage (not accessible from other files) or external linkage (can be accessed by other files) ?

- What is the purpose of each field between the parenthesis in a for loop ? for (a;b;c)…

- What is the difference between :
while (j > 0) do {…}
  and
do {…} while (j>0)

Review Questions

- Use bit operators to implement a function which returns TRUE if a positive number x is odd.
  Remember: TRUE = non_zero, FALSE=0

```
int is_odd(int x)
{
        /* some binary operation */

}
```

Find the Errors

-
  ```
  #include <stdio.h>
  main() {
      float f;
      scanf(“%d”,&f);
      switch (square(g)) {
        case ‘4’:
                   printf(“Got a 2\n”);
                   break;
        case 16:
                   switch(f) {
                     case 1:
                           printf(“Got a 1\n”);
                     case 2:
                           printf(“Got a 2\n”);
                           break;
                     default:
                           printf(“nothing..\n”);
                   }
      }
  }
  int g = 4;
  int square(int j) { return (j*j); }
  ```

Array Review Questions

- Assume:
  - **fixed** is a 2D array of integers
  - the actual address of **fixed** is "1000"
  - integers are 4 bytes long
- For each line, answer:
  1) Is it legal (compiles)?
  2) Can it cause a run-time error ?
  3) If it runs, what is the outcome ?

```
…
int fixed[30][40];
printf("%d ",&(fixed[0][0]));   /* line 1*/
printf("%d ",&(fixed[0][30])); /* line 2*/
printf("%d ",&(fixed[0][45])); /* line 3*/
printf("%d ",&(fixed[1][5]));   /* line 4*/
printf("%d ",&(fixed[30][10])); /* line 5*/
fixed[0][10] = 10;               /* line 6*/
fixed[0][10] = 'a';              /* line 7*/
fixed[30][0] = 15;              /* line 8*/
```

What is the output ?

-
  ```
  int i=1;
  int reset(){   return (i);  }
  int next(int j) { return (j = i++;);  }
  int last(int j) {
          static int i = 10;
          return (j = i--);
  }
  int new(int i) { int j=10;  return i = (j += i) ;  }
  main()
  {
          int i, j;
          i = reset();
          for (j=1;j<=3;j++)
          {
               printf(“”%d %d\n”,i,j);
              printf(“%d\n,next(i));
              printf(“%d\n,last(i));
              printf(“%d\n,new(i+j));
          }
  }
  ```

What is the output ?

- #define N 1000

  ```
  main()
  {
    int i,j,a[N];

    for (i=2; i<N;i++)
         a[i] = 1;
    for (i=2; i< N; i++)
         if (a[i])
            for (j = i; j < N/i; j++)
                a[i*j] = 0;
     for (i=2; i< N; i++)
         if (a[i])
            printf(“%d “,i);
     printf(“\n”);
  }
  ```