

강의명: 프로그래밍

실습 번호: 4

실습 제목: Control Flow (제어 흐름)

학생 이름: 임지빈

학번: 202211051

1. 섭씨 화씨 변환 (for 문 사용)

1.1

```
for (c=0; c <= 50; c=c+10) {  
    f = c * (9.0/5.0) + 32;  
    printf("C=%d , F=%d\n" , c , f);  
}
```

C가 0부터 시작해서 50이하일때까지 f를 출력하는 함수

1.2

5번, c가 10 커졌다

1.3

```
s2211051@oak:lab04$ emacs c2f-for.c  
s2211051@oak:lab04$ gcc c2f-for.c -o c2f-for  
s2211051@oak:lab04$ ./c2f-for  
C=0 , F=32  
C=10 , F=50  
C=20 , F=68  
C=30 , F=86  
C=40 , F=104  
C=50 , F=122
```

2. 섭씨 화씨 변환 (while 문 사용)

2.1

```
c=0;  
while(c <= 50) {  
    f = c * (9.0/5.0) + 32;  
    c=c+10;  
    printf("C=%d , F=%d\n" , c , f);  
}
```

C 가 50 이하일때 계속 반복하는 함수

2.2

```
s2211051@oak:lab04$ emacs c2f-while.c
s2211051@oak:lab04$ gcc c2f-while.c -o c2f-while
s2211051@oak:lab04$ ./c2f-while
C=10 , F=32
C=20 , F=50
C=30 , F=68
C=40 , F=86
C=50 , F=104
C=60 , F=122
s2211051@oak:lab04$
```

3. 섭씨를 화씨로 변화 (do 문 사용)

3.1

```
c=0;
do {
    f= c* (9.0/5.0) + 32;
    printf ("C=%d, F=%d\n" , c , f);
    c=c+10;
} while
(c<= 50);
```

C 가 50 이하일때 계속 반복하는 함수

3.2

```
s2211051@oak:lab04$ gcc c2f-do.c -o c2f-do
s2211051@oak:lab04$ ./c2f-do
C=0, F=32
C=10, F=50
C=20, F=68
C=30, F=86
C=40, F=104
C=50, F=122
```

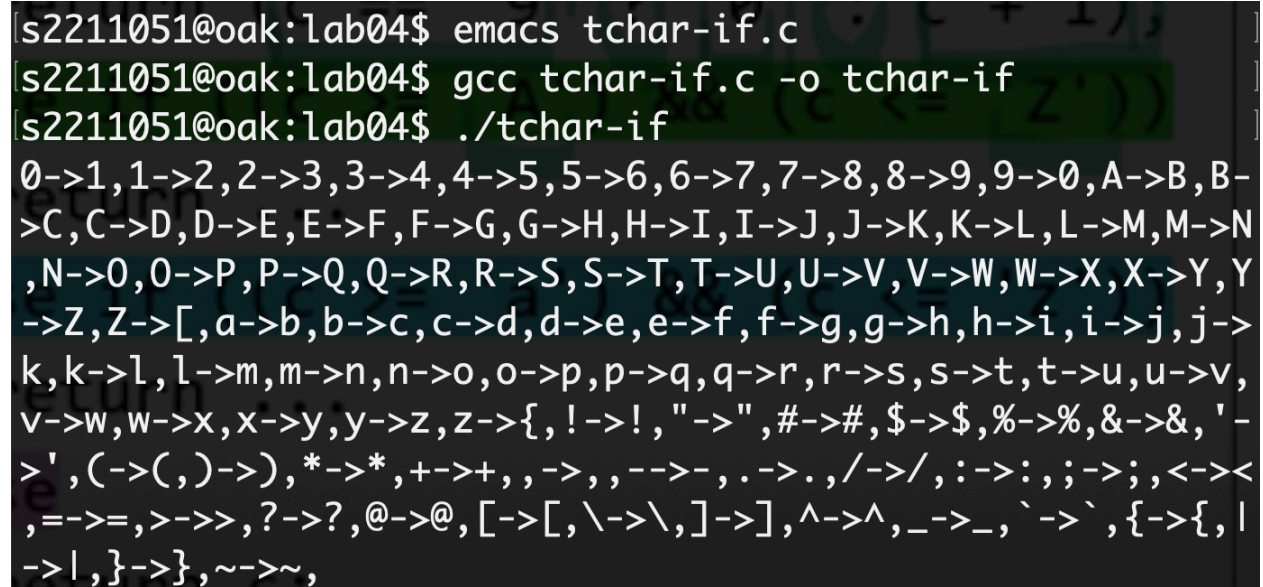
4. 문자 변환 (if-else-if 문 사용)

4.1

```
if ((c >= '0') && (c <= '9'))
    return (c == '9' ? '0' : c+1);
else if ((c >= 'A') && (c <= 'Z'))
    return(c == '9' ? '0' : c+1);
else if ((c>= 'a') && (c <= 'z'))
    return(c == '9' ? '0' : c+1);
else
    return c;
```

c 가 숫자일때, 소문자일때, 대문자일때, 나머지 문자들일때로 else if 를 쓰는 함수

4.2



```
s2211051@oak:lab04$ emacs tchar-if.c
s2211051@oak:lab04$ gcc tchar-if.c -o tchar-if
s2211051@oak:lab04$ ./tchar-if
0->1,1->2,2->3,3->4,4->5,5->6,6->7,7->8,8->9,9->0,A->B,B->C,C->D,D->E,E->F,F->G,G->H,H->I,I->J,J->K,K->L,L->M,M->N,N->O,O->P,P->Q,Q->R,R->S,S->T,T->U,U->V,V->W,W->X,X->Y,Y->Z,Z->[,a->b,b->c,c->d,d->e,e->f,f->g,g->h,h->i,i->j,j->k,k->l,l->m,m->n,n->o,o->p,p->q,q->r,r->s,s->t,t->u,u->v,v->w,w->x,x->y,y->z,z->{,!->!,\"->\",#->#,$->$,%->%,&->&,'->',(->(,) ->), *->*,+->+, , -> , , --> -, . -> . , / -> / , :-> :, ; -> ; , <-> < , =-> = , >-> > , ?-> ? , @-> @ , [-> [ , \ -> \ , ] -> ] , ^-> ^ , _-> _ , ` -> ` , {-> { , | -> | , } -> } , ~ -> ~ ,
```

5. 문자 변환 (switch 문 사용)

5.1

5.2

5.3

6. 좌표 출력 (for 문 사용)

6.1

```
for (x=0; x<5; x++) {
```

```

    for (y=0; y<10; y++) {
        printf("(%d,%d)", x, y);
    }
    printf("\n");
}

```

X 는 5 이하, y 는 10 이하인 함수를 출력

6.2

```

s2211051@oak:lab04$ gcc xy-for.c -o xy-for
s2211051@oak:lab04$ ./xy-for
(0,0)(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7)(0,8)(0,9)
(1,0)(1,1)(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)(1,8)(1,9)
(2,0)(2,1)(2,2)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)(2,9)
(3,0)(3,1)(3,2)(3,3)(3,4)(3,5)(3,6)(3,7)(3,8)(3,9)
(4,0)(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)(4,7)(4,8)(4,9)
s2211051@oak:lab04$

```

7.좌표 출력 (for 및 break 문 사용)

7.1

```

for (x=0; x<5; x++) {
    for (y=0; y<10; y++) {
        printf("(%d,%d)", x, y);
    }
    if (y==9) break;
    printf("\n");
}

```

Y 가 9 일때 break 하는 함수

7.2

```

s2211051@oak:lab04$ gcc xy-break.c -o xy-break
s2211051@oak:lab04$ ./xy-break
(0,0)(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7)(0,8)(0,9)
(1,0)(1,1)(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)(1,8)(1,9)
(2,0)(2,1)(2,2)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)(2,9)
(3,0)(3,1)(3,2)(3,3)(3,4)(3,5)(3,6)(3,7)(3,8)(3,9)
(4,0)(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)(4,7)(4,8)(4,9)
s2211051@oak:lab04$

```

8.좌표 출력 (for 및 continue 문 사용)

8.1

```
for (x=0; x<5; x++) {  
    for (y=0; y<10; y++) {  
        printf("(%d,%d)", x, y);  
        if (x<5) continue;  
    }  
    printf("\n");  
}
```

X 가 5 이하일때 계속 하는 함수

8.2

```
s2211051@oak:lab04$ emacs xy-continue.c  
s2211051@oak:lab04$ gcc xy-continue.c -o xy-continue  
s2211051@oak:lab04$ ./xy-continue  
(0,0)(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7)(0,8)(0,9)  
(1,0)(1,1)(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)(1,8)(1,9)  
(2,0)(2,1)(2,2)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)(2,9)  
(3,0)(3,1)(3,2)(3,3)(3,4)(3,5)(3,6)(3,7)(3,8)(3,9)  
(4,0)(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)(4,7)(4,8)(4,9)  
s2211051@oak:lab04$
```