Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – 장성환
redmk1025@gmail.com

후위연산 변환 연습

```
#include <stdio.h>

char stack[50], polish[50];
int pri[256];
int sp1, sp2;

int main(void){
   int i;
   char *p = "a+b-c*d/e+3*2";

   for(i=0; i<256; i++){
      pri[i] = 3;
   }
   pri['+']=pri['-']=1;
   pri['*']=pri['/']=2;


   stack[0]=0;
   pri[0]=-1;

   sp1 = sp2 = 0;

   while(*p!='\0'){
      while(pri[*p] <= pri[stack[sp1]]){
         polish[++sp2] = stack[sp1--];
      }
      stack[++sp1] = *p++;
   }
```

```
  for(i=sp1; i>0; i--){
    polish[++sp2] = stack[i];
  }

  for(i=1; i<sp2; i++){
    putchar(polish[i]);
  }
  printf("\n");

  for(i=0; i<sp2; i++){
    printf("%d\n",polish[i]);
  }

  return 0;
}
```

미완

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

typedef struct _stack{
  char data[50];
  struct _stack* link;
}stack;

stack* get_stk_mem();
void push(stack **sp, char * data);
char* pop(stack **sp);
void print_stack(stack *sp);
```

```
void prim_init(char (*pri)[]);
char check(stack *p);

int main(void){
   int i=0;
   char pri[256];
   char p[100];

   stack *stack_sp = NULL;
   stack *polish_sp = NULL;

   prim_init(&pri);
   scanf("%s",p);

   return 0;
}

void prim_init(char (*pri)[]){
   int i;
   for(i=0; i<255; i++){
      (*pri)[i]= 5;
   }
   (*pri)['+']=2;(*pri)['-']=2;
   (*pri)['*']=3;(*pri)['/']=3;
   (*pri)['(']=6;(*pri)[')']=0;
   (*pri)['^']=4;(*pri)['l']=4;
}

stack* get_stk_mem(){
   stack *tmp=(stack*)malloc(sizeof(stack));
   tmp->link=NULL;
```

```
   return tmp;
}

void push(stack **sp, char * data){
   stack *tmp = *sp;
   (*sp) = get_stk_mem();
   sprintf((*sp)->data,"%s", data);
   (*sp)->link = tmp;
}

char* pop(stack **sp){
   stack *tmp = *sp;
   char *tmp_data  = tmp->data;
   (*sp)=tmp->link;
   free(tmp);
   return tmp_data;
}

char check(stack *p){
   if(p==NULL){
      return 0;
   }
   char * tmp = p->data;
   return tmp[0];
}

void print_stack(stack *sp){
  if(sp==NULL)
     return;
   printf("%s ",sp->data);
   print_stack(sp->link);
}
```