24RoboMaster醒狮电控组梯队考核

> ----------C语言篇

PART I

Q1.以下哪种不是C语言的基本数据类型？
A. char　　B. int　　C.short　　D.num

Q2.以下哪个是双精度浮点型？
A. int　　B.double　　C.num　　D.float

Q3.下列哪个是正确的赋值方式？
A.boss==1;　　B.boy>>1;　　C.girl=1;　　D.you&&1;

Q4.下列哪个是错误的运算符？
A.&&　　B.||　　C.??　　D.!=

Q5. 请问以下关于printf的表达式正确的是
A. printf(“this is C program.”\n);
B. printf(‘%c’a);
C. printf(‘%5.2f’,3.1415926);
D. printf(“\n%12.3e”,12346.6788);

Q6.判断:在复合逻辑运算符中,x+=y+1相当于y=x+y+1?
A.正确
B.错误

Q7.在 int a=5,y=20; a+=(y%(5+a)*a)/y; 中，若没有逻辑错误，a的值为？

Q8.C语言while和do-while循环的主要区别是？
A.while循环的控制条件比do-while循环控制条件严格
B.do-while的循环体至少无条件执行一次
C.do-while允许从外部转到循环体内
D.do-while的循环体不能是复合语句

Q9.下列有关for循环的正确描述是?
A.for循环只能用于循环次数已经确定的情况
B.for循环是先执行循环体语句后判定表达式
C.在for循环中，不能用break语句跳出循环体
D.for循环体语句中可以包含多条语句，但要用花括号括起来

Q10.break语句和continue语句的区别是什么？

PART II

Q11:举例出C语言中的数据类型（越多越好）

```
eg:
int ;
float ;
enum ;
......
```

Q12:使用 `while` 自定义一个死循环函数

```
//
```

Q13：解释 `VAL_LIMIT` ，并计算 `Alpha` 、 `Beta`、 `Gamma` 的最终值

```
#define MINVALUE -19
#define MAXVALUE 43
#define VAL_LIMIT(val, min, max) \
do {\
if((val) <= (min))\
{\
  (val) = (min);\
}\
else if((val) >= (max))\
{\
  (val) = (max);\
}\
} while(0)\

float Alpha = 1.0;
float Beta = 76.4;
float Gamma = -57.3;

VAL_LIMIT(Alpha,MINVALUE,MAXVALUE);
VAL_LIMIT(Beta,MINVALUE,MAXVALUE);
VAL_LIMIT(Gamma,MINVALUE,MAXVALUE);
```

Q14:于 `笛卡尔坐标系` 中表示 `now1` 与 `now2` 的 `变化曲线`

```
float now1 = 50.5 ;
float now2 = -30.3 ;

float ref1 = 100.1;
float ref2 = -70.7;

float high = 1.0 ;
float low = 1.0 ;

void buffer(float *a, float b, float high_parameter, float low_parameter)
```

```
{
  if (((*a - b) <= high_parameter) && ((*a - b) >= -low_parameter))
  {
    *a = b;
  }
  else
  {
    if (*a < b)
      *a += high_parameter;
    if (*a > b)
      *a -= low_parameter;
  }
}

buffer(&now1,ref1,high,low);
buffer(&now2,ref2,high,low);
```

Q15:请自定义变量，并巧妙利用 ramp_t，ramp_init，ramp_calc 实现自定义变量的斜坡式下降与斜坡式增长

```
typedef struct
{
  uint32_t get_count;
  uint32_t set_count;
  float out;
}ramp_t;

void ramp_init(ramp_t *ramp,uint32_t target_count)
{
    ramp->get_count = 0;
    ramp->set_count = target_count;
    ramp->out = 0;
}

float ramp_calc(ramp_t *ramp)
{
  if(ramp->set_count <= 0)
    return 0;
  if(ramp->get_count >= ramp->set_count)
    ramp->get_count = ramp->set_count;
  else
    ramp->get_count++;

  ramp->out = (float)ramp->get_count/(float)ramp->set_count;
  return ramp->out;
}
```

Q16:现有名为 HAL_GetTick 的时间测量函数,其单位为 ms,请利用 HAL_GetTick 函数返回值的特殊性质,实现程序1ms 定时打印"Hello,world !"`(tip:① 定义变量与函数配合获得Δt值; ② if语句)

```
//返回值 uwTick 为当前系统运行时间
float HAL_GetTick(void)
{
  return uwTick;
}
```

Q17:使用名为 callback 的 函数指针 初始化 alpha与 beta 的值

```
#define  VALUE_B  217
#define  VALUE_C  219.321

static int alpha;
static float beta;

void STDRxCallback(int* b, float* c)
{
    *b = VALUE_B;
    *c = VALUE_C;
}
void (*callback)(int* c, float* b);
```

Q18:请你初始化 alpha的值为gama 的低八位、 beta的值为 gama 的高八位

```
注：uint8_t 为无符号8位整型
    uint16_t 为无符号32位整型

uint8_t alpha = 0;
uint8_t beta = 0;

uint16_t gama = 20001;
```

△Q19:下图代码为PID控制算法公式,请 `自定义结构体` ,并利用 `pid_t`、`PID_Struct_Init`、`pid_calc` 分别初始化 `kp`、`ki`、`kd`、`maxout`、`integral_limit` 的值，并得到经 `pid_calc` 计算后的 `最终值` (tip: 可令 kp = 1 ,ki = 0.01,kd = 20 ,maxout = 500,integral_limit = 100 并将其传入 PID_Struct_Init 进行初始化,最后调用 pid_calc 得到最终值)`

```
enum
{
    NOW_ERR = 0,
    LAST_ERR,
    LLAST_ERR,
};

typedef struct pid
{
    float set;
    float get;
    float error[3];

    float kp;
    float ki;
    float kd;

    float pout;
    float iout;
    float dout;
    float out;


    int32_t maxout;
    int32_t integral_limit;

    void (*f_pid_init)(struct pid *pid_t,
                                        float p,
                                        float i,
                                        float d,
                                        int32_t max_out,
                                        int32_t integral_limit);
    void (*f_pid_reset)(struct pid *pid_t,
                                        float p,
                                        float i,
                                        float d);

}pid_t;


static void abs_limit(float *x,int32_t limit)
{
    if(*x > limit)
        *x = limit;
    if(*x < -limit)
```

```
        *x = -limit;
}

static void pid_init(pid_t *pid,float p,float i,float d,int32_t max_out,int32_t
integral_limit)
{
    pid->kp = p;
    pid->ki = i;
    pid->kd = d;
    pid->maxout = max_out;
    pid->integral_limit = integral_limit;
}

static void pid_reset(pid_t *pid,float p,float i,float d)
{
    pid->kp = p;
    pid->ki = i;
    pid->kd = d;

    pid->pout = 0;
    pid->iout = 0;
    pid->dout = 0;
    pid->out  = 0;
}

void PID_Struct_Init(pid_t *pid,float p,float i,float d,int32_t max_out,int32_t
integral_limit)
{
        pid->f_pid_init = pid_init;
        pid->f_pid_reset = pid_reset;

        pid->f_pid_init(pid,p,i,d,max_out,integral_limit);
        pid->f_pid_reset(pid,p,i,d);
}

float pid_calc(pid_t *pid,float get,float set)
{
    pid->get = get;
    pid->set = set;
    pid->error[NOW_ERR] = set - get;

      pid->pout = pid->kp * pid->error[NOW_ERR];
      pid->iout += pid->ki * pid->error[NOW_ERR];
      pid->dout = pid->kd * (pid->error[NOW_ERR] - pid->error[LAST_ERR]);
      //积分限幅
      abs_limit(&(pid->iout),pid->integral_limit);
      pid->out = pid->pout + pid->iout + pid->dout;
      abs_limit(&(pid->out),pid->maxout);
      //最终值限幅
      pid->out += pid->pout + pid->iout + pid->dout;
      abs_limit(&(pid->out), pid->maxout);

  pid->error[LLAST_ERR] = pid->error[LAST_ERR];
  pid->error[LAST_ERR]  = pid->error[NOW_ERR];
  //最终值输出
    return pid->out;
```

```
}
```

PART III

Q20：题目：请自定义一个结构体类型"Student"，包含以下成员变量（特殊要求：用结构体与指针相关内容解决）

```
姓名（字符串类型）
学号（整型）
成绩（单精度浮点型）
请编写一个函数，接收一个指向 "Student" 结构体数组的指针，将每个学生的姓名、学号和成绩输出到屏幕上。
```

Q21：请编写一个函数，接收一个整型数组和数组长度作为参数，找出数组中的最大值，并通过指针将最大值返回给调用函数。（特殊要求:利用结构体与指针相关内容解决）

```c
#include <stdio.h>
int main() {
    int arr[] = {8, 11, 4, 2, 6};
    int length = sizeof(arr) / sizeof(arr[0]);
    int max;
  //在此填写函数 ↓

  //在此填写函数 ↑
    printf("最大值：%d\n", max);
    return 0;

>注意：已给代码不可更改!!!
```

Q22：请编写一个void函数，使其限制电机转速在-4000~4000范围内（特殊要求:在不用全局变量的情况下完成 ）

```c
#include <stdio.h>
> 注意：不准在这给变量赋值，只可给函数声明!
int main() {
    int nowspeed=5000,maxspeed=4000;
    //请插入一个void函数 ↓

    //请在此插入一个函数 ↑
    printf("nowspeed的值",nowspeed)
    return 0;
}
//请在此补充函数↓
void ...
//请在此补充函数↑

> 注意:①不可加入全局变量!
```

> ②已给代码不可更改!

Q23：已知 *云台* 有四种状态，分别为 *GIMBAL_INIT_NEVER*，*GIMBAL_INIT_DONE*，*NO_ACTION*，*IS_ACTION* 四种，请你定义一个 *枚举结构体gimbal_state_t* 放入这四种状态；假设 *现在* 云台状态为 *GIMBAL_INIT_DONE*，请你写出程序,使其输出 *当前状态* 的枚举值 *（特殊要求：利用枚举）*

```
//
```

PART IV

Q24：请定义一个 *枚举类型"CourseType"*，包含以下选项：*数学*、*英语*、*物理*、*化学*。定义一个 *结构体类型"Course"*，包含以下成员：*课程类型（type）：枚举类型"CourseType"*；*学分（credit）：整数类型*；*成绩（score）：浮点数类型* 。并声明一个 *指向"Course"结构体的指针"courses"* 。编写一个 *函数"addCourse()"*，该函数接收用户输入的 *课程类型、学分和成绩*，并将其添加到 *"courses"所指向的结构体数组* 中。如果数组已 *满*(组数为10)，则提示 *错误信息*。编写一个 *函数"calculateGPA()"*，该函数接收 *"courses"指针和数组长度作为参数*，计算所有课程的平均学分绩点（GPA）。学分绩点按照以下规则计算：90-100分为A，4.0绩点；80-89分为B，3.0绩点；70-79分为C，2.0绩点；60-69分为D，1.0绩点；低于60分为F，0绩点。在主函数中，首先让用户输入课程数量，调用"addCourse()"函数让用户依次输入课程信息，最后 *调用"calculateGPA()"函数* 计算 *平均学分绩点* 并 *打印* 出来。

```
//
```