### Day07

1、指出以下代码片段的不妥之处?

struct node

{

char itable;

int num[20];

char \* togs;

};

int x;

char\* togs在struct node模板里,不是野指针

不妥之处是int x没放入结构体模板

2、设计一个结构模板，保存一个月份名、一个3个字母的该月份的缩写、该月的天数，以及月份号。

#include <stdio.h>

#define MAXSIZE 20

struct month

{

char name[MAXSIZE];

char simple\_month[MAXSIZE];

int day;

int number;

};

int main(int argc, char const \*argv[])

{

struct month month = {0};

int i=0;

gets(month.name);

for(i=0;i<3;i++)

{

month.simple\_month[i] = month.name[i];

}

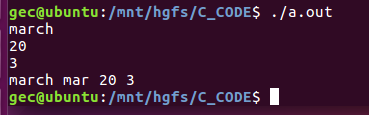
scanf("%d",&(month.day));

scanf("%d",&(month.number));

printf("%s %s %d %d\n", month.name, month.simple\_month, month.day, month.number);

return 0;

}



3、分析以下结构所占的存储空间大小:

struct animals

{

char dog;

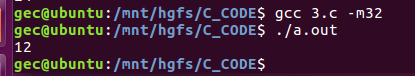
unsigned long cat;

unsigned short pig;

char fox;

};

12



4、定义一个结构体变量(包括年月日)。计算该日在本年中是第几天?注意闰年问题。

#include <stdio.h>

struct data

{

int year;

int month;

int day;

};

int is\_leap(int year)

{

if (year % 4 == 0)

{

if(year % 100 == 0)

{

return 1;

}

else

{

return 0;

}

}

else if (year % 400 == 0)

{

return 1;

}

else

{

return 0;

}

}

int main(int argc, char const \*argv[])

{

struct data data1 = {0};

int sum = 0;

struct data\* p = &data1;

scanf("%d%d%d", &(data1.year), &(data1.month), &(data1.day));

if(is\_leap(data1.year))

{

sum += 1;

}

switch (data1.month-1)

{

case 12: sum +=31;

case 11: sum +=30;

case 10: sum +=31;

case 9: sum +=30;

case 8: sum +=31;

case 7: sum +=31;

case 6: sum +=30;

case 5: sum +=31;

case 4: sum +=30;

case 3: sum +=31;

case 2: sum +=28;

case 1: sum +=31;break;

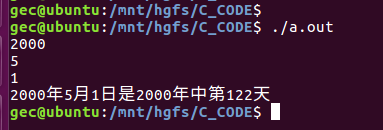
}

printf("%d年%d月%d日是%d年中第%d天\n", data1.year, data1.month, data1.day

, data1.year, sum+data1.day);

return 0;

}



5、声明一个枚举类型，使用choices作为标记，将枚举常量no、yes和maybe分别设置为0、1和2。

#include <stdio.h>

enum choice{

yes,

no,

maybe

};

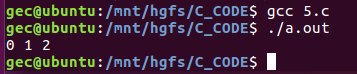
int main(int argc, char const \*argv[])

{

printf("%d %d %d\n", yes, no, maybe);

return 0;

}



6、声明4个函数，并把一个指针数组初始化为指向它们。每个函数接受两个double参数并返回double值。

#include <stdio.h>

double add(double a, double b)

{

return a+b;

}

double subtraction(double a, double b)

{

return a-b;

}

double multiplication(double a, double b)

{

return a\*b;

}

double division(double a, double b)

{

if (b == 0)

{

printf("error!\n");

return 0;

}

return a/b+(int)a%(int)b;

}

int main()

{

double (\*calc[4])(double, double) = { add, subtraction

, multiplication, division};

double a, b;

scanf("%lf%lf", &a, &b);

int i = 0;

for (i=0;i<4;i++)

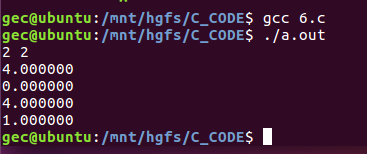
{

printf("%lf\n", calc[i](a, b));

}

return 0;

}



7、假设有以下说明和定义:

typedef union

{

long i;

int k[5];

char C; .

} fruit; //20

struct creature

{

short cat;

fruit apple;

double dog;

}; //32

fruit berry;

则语句printf("%d"，sizeof(struct creature)+sizeof(berry));的执行结果是?

52

8、编写一个transform()函数，它接受4个参数:包含double类型数据的源数组名, double类型的目标数组名,表示数组元素个数的int变量以及一个函数名(或者等价的指向函数的指针)。transform()函数把指定的函数作用于源数组的每个元素，并将返回值放到目标数组中。

例如: transform(source， target, 100， sin);这个函数调用sin(source[0])赋给target[0]， 等等。共有100个元素。在一个程序中测试该函数，调用4次transform()，分别使用math.h函数库中的两个函数以及自己设计的两个适合的函数作为参数。

#include <stdio.h>

#include <math.h>

#define MAXSIZE 3

void transform(double source[], double target[], int n, double (\*fun)(double))

{

int i=0;

for (i=0;i<n;i++)

{

target[i] = fun(source[i]);

printf("%lf\n", target[i]);

}

printf("============================\n");

}

int main()

{

double src[MAXSIZE]={1, 2, 3};

double tar[MAXSIZE]={0};

transform( src, tar, MAXSIZE, &sin);

transform( src, tar, MAXSIZE, &cos);

transform( src, tar, MAXSIZE, &tan);

transform( src, tar, MAXSIZE, &sqrt);

return 0;

}

