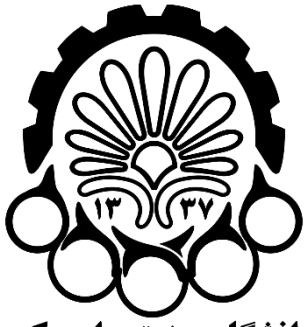
"به نام دادار دادگستر"



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

کارگاه مبانی کامپیوتر و برنامه نویسی

عنوان

تمرین دهم (C-Lab_pointer)

مدرس
مهندس امیرحسین بابائیان

دانشجو
محمد یاراحمدی
۴۰۲۳۱۰۵۹

ترم پاییز ۰۳ - ۰۲
دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

فهرست

پرسش ۱

برای یادگیری عملکرد پوینترها لازم است که روی تحلیل برنامه های پوینتری مسلط باشیم و بتوانیم آ نها را دیباگ کنیم. بیایید برای شروع با قطعه کدهای کوتاه و نکته دار شروع کنیم تا دانش پوینتری مان را محک بزنیم و نکات را یکی یکی بررسی کنیم. قطعه کد زیر را اجرا کنید. در خروجی چه چیزی مشاهده می کنید؟ برنامه را خط به خط تحلیل کنید و توضیح دهید که چطور این خروجی حاصل شده است؟.

```c
#include <stdio.h>
//soal1
int main() {
    int a[5] = {1 , 2 , 3 , 4 , 5} ;
    printf("%lu\n" , sizeof(a)) ;
    int *ptr = (int *)(&a + 1) ;
    printf("%d %d\n" , *(a + 1) ,*(ptr - 1));
    return 0 ;
}
```

پاسخ پرسش ۱

در ابتدا یک آرایه ۵ عضوی به ما داده شده است که هر عضو آرایه خود متغیری از جنس int می باشد . پس وقتی که سایز آرایه را می خواهیم چاپ کنیم در خروجی ترمینال ، عدد ٤*٥ ظاهر خواهد شد . int *ptr به یک بیت بعد از آرایه اشاره میکند . بعبارتی به بیت ۲۱ ام پس از آدرس اولین خانه آرایه اشاره میکند که وقتی عبارت *(ptr-۱) را میخواهیم پرینت کنیم به بیت ۲۰ ام پس از خانه اول اشاره میکند که این به معنای آخرین عضو آرایه می باشد . همچنین *(a+۱)=a[۱] می باشد .

خروجی ترمینال این قطعه کد را مشاهده می کنید .

```
۲۰
۲ ٥
```

پرسش ۲

حال سعی کنید بدون چاپ کردن خروجی نهایی این قطعه کد ، آن را تحلیل کنید و تشخیص دهید که خروجی چه خواهد بود.

```c
#include <stdio.h>
void f(char **);

int main() {
    char *argv[] = {"ab", "cd", "ef", "gh", "ij", "kl"};
    f(argv) ;
    return 0;
}

void f(char **p){
char *t;
t = (p += sizeof(int))[-1];
printf("%s" , t );
}
```

پاسخ پرسش ۲

در این قطعه کد ، تابع f را داریم که در آن ، با توجه به سایز متغیر int که ٤ است طبق دستور

p += sizeof(int)[-١] عـدد ١ – ٤ میشـود و پـوینتر p بـه ایـن آدرس اشـاره میکنـد .(p[٣]) . در تـابع main وقتـی ایـن تـابع را صدا میزنیم عضو سوم این آرایه argv پرینت خواهد شد.

```
gh
```

پرسش ۳

حال سعی کنید کمی عمیقتر و دقیقتر کدها را بررسی کنید. خطای برنامه زیر را پیدا کرده و آن را اصلاح کنید. به نظر شما برنامه ی فعلی درست کار می کند؟ فکر می کنید علت این اتفاق چیست؟

```
#include <stdio.h>
int f(int* p){
    printf("a = %d\n" , *p) ; // a=10?
}
    int main(){
    int a = 10 ;
    f((int *)a);
}
```

پاسخ پرسش ۳

هنگامیکه قصد فراخوانی تابعی را داریم که در پروتایپ آن ، اشاره گر ورودی میگیرد ، باید آدرس آن متغیر را به تابع پاس دهیم .

در این قطعه کد اشتباها خود متغیر به تابع داده شده است . شکل درست این قطعه کد به صورت زیر است :

```
#include <stdio.h>
int f(int* p){
    printf("a = %d\n" , *p) ; // a=10?
}
    int main(){
    int a = 10 ;
    f(&a);
}
```

پرسش ٤

تفاوت متغیرهای a و b در چیست؟

```
int *a[3];
int (*b)[3] ;
```

خروجی قطعه کد زیر چیست؟

```
#include <stdio.h>

int *a[3] ;
int (*b)[3] ;

int main(){
    int a[][3] = {1 , 2 , 3 , 4 , 5 , 6 } ;
    int (*ptr)[3] = a ;
    printf("%d %d", (*ptr)[1] , (*ptr)[2]) ;
    ++ptr ;
    printf("%d %d\n" , (*ptr)[1] , (*ptr)[2]);
    return 0 ;
}
```

پاسخ پرسش ٤

این ۲ عبارت در حافظه به صورت متفاوتی قرار میگیرند . int *a[۳] سه خانه حافظه را اشغال میکند که هر کدام یک اشاره گر به یک متغیر int هستند. int(*b)[۳] یک خانه حافظه را اشغال میکند که یک اشاره گر به اولین خانه یک آرایه سه تایی از int است . ptr پوینتری است که به a[۰] اشاره می کند . پس (*ptr)[۱] و (*ptr)[۲] به ترتیب به a[۱] و a[۲] اشاره می کنند .

اگر ptr یک واحد اضافه شود به سه جمله دوم آرایه اشاره خواهد کرد .

```
 2 3 5 6
```

پرسش ۵

فکر می کنید چرا کد زیر به درستی ۵ فاکتوریل را حساب نمی کند؟

```
void factorial (int *res , int num ) {
    *res =1 ;
    for (int i = 1 ; i <= num ; i++){
        *res *= i ;
    }
}

int main(){
    int *res;
    int num = 5 ;
    factorial(*res , num ) ;
    printf("%d! = %d" , num , *res) ;
    return 0 ;
}
```

پاسخ پرسش ۵

باید آدرس متغیر به تابع پاس داده شود که قطعه خود اشاره گر به آن متغیر را به تابع فراحوانی شده ، داده است . همچنین به جای اشاره گر به res باید مقدار خود متغیر داده می شد چون تابع مقدار متغیر را عوض میکند . خروجی ترمینال و شکل صحیح کد به صورت زیر است :

```
۵! = ۱۲۰
```

```
void factorial (int *res , int num ) {
    *res =1 ;
    for (int i = 1 ; i <= num ; i++){
        *res *= i ;
    }
}

int main(){
    int res;
    int num = 5 ;
    factorial(&res , num ) ;
    printf("%d! = %d" , num , res) ;
    return 0 ;
}
```