به نام خدا

دستور کار کارگاه مبانی کامپیوتر و برنامه‌نویسی

جلسه یازدهم

اشاره‌گرها قسمت اول

۱. خروجی قطعه کد زیر چیست؟

```c
#include <stdio.h>

void fun(int *p) {
  int q = 10;
  p = &q;
}

int main() {
  int r = 20;
  int *p = &r;
  fun(p);
  printf("%d\n", *p);
  return 0;
}
```

۲. خروجی قطعه کد زیر چیست؟

```c
#include <stdio.h>

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;
}
```

۳. خروجی قطعه کد زیر چیست؟

```c
#include <stdio.h>

// Assume that the size of int is 4.

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("%sn", t);
}
```

۴. فرض کنید جدول زیر نمایی از حافظه باشد:

Address	Value
1	20
2	‘H’
3	‘I’
4	10.2
5	‘S’
6	0

در این صورت خروجی قطعه کد زیر چه خواهد بود؟

```
int *p = 1;
int *q = p;
*p = 20
*q++;
printf("%d\n", *p);
q++;
pritnf("%d\n", *q);
p = 5;
printf("%s\n", p);
p = 2
printf("%s\n", p);
```

راهنمایی: می‌دانید که رشته‌ها در C می‌بایست Null Terminated باشند از این مفهوم برای کشف یک خطای زمان اجرا در کد فوق استفاده کنید.

۵. خطای برنامه زیر را پیدا کرده و آن را اصلاح کنید. به نظر شما برنامه‌ی فعلی (دارای خطا) درست کار می‌کند؟ فکر می‌کنید علت این اتفاق چیست؟

```c
#include <stdio.h>

int f(int* p) {
  printf("a = %d\n", p); // a = 10?
}

int main() {
  int a = 10;
  f((int *) a);
}
```

۶. تفاوت متغیرهای a و b در چیست؟

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

خروجی قطعه کد زیر چیست؟

```c
#include <stdio.h>

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;
}
```

۷. در این قسمت قصد داریم شما را با چگونگی پیاده‌سازی عملیات روی اشاره‌گرها آشنا کنیم. در عمل این عملیات‌ها بر اساس اندازه متغیری که اشاره‌گر به آن‌ها اشاره می‌کند کار می‌کنند. مثلا

```c
int *a;
a++;
```

اشاره‌گر a را به اندازه‌ی sizeof(int) جلو می‌برد.

برای دیدن امر، خروجی کد C را به زبان اسمبلی تبدیل می‌کنیم:

```c
int main() {
  int *p = 0;
  p++;
}
```

```
    movq    %rsp, %rbp
    xorl    %eax, %eax
    movq    $0, -8(%rbp)
    movq    -8(%rbp), %rcx
    addq    $4, %rcx
    movq    %rcx, -8(%rbp)
```

همانطور که در کد اسمبلی فوق مشخص است، مقدار افزایش برابر با عدد ۴ می‌باشد که اندازه یک متغیر int به بایت می‌باشد.