

سؤال ۱ زبان ترکی به مفعول قرار دارد low-level یا سطح زبان programming language

اسمها یا زبان ماشین و ماحدی طوا . این زبان ها سینتکس و گرامر

خوب قابل خواندن است این است یعنی گرامرهای ترکی زبان خود کامپیوتر

مفعول قرار است و آنجا که گرامرهای این زبان ها مفعول است

چون مستقیماً می تواند به مفعول قرار دهد و نیازی به ترجمه نیست

کامپیوتر نیستند . گرامرهای زبان ها به این شروع گفته های به نام نویسی

مفعول است و کار با آن علم محلی تری می خواهد در موردی که حافظه کم تر

استفاده کرد ۶۰۰۰ به ربع تر انعام می شود SAHEL

زبان C++ دارای بینایی و مرسوم low-level کاری است و دارای این است:

این درستی دارد که با پوینترها که مستقیماً با حافظه و کار با آن مربوط است، پوینتر

این ها low-level است.

```
int * ptr;
```

```
ptr = new int;
```

```
delete ptr;
```

در یک pointer فارسی low-level C++، این است که از smart pointer استفاده

استفاده می شود که با پوینترها از جمله مدیریت حافظه و سایر مفاهیم low-level language

connections را را می بیند. استفاده از اشاره از raw pointer ها می تواند

منجر به memory leak از آن ها می شود که با آن ها می شود.

smart pointer ها از این پوینترها deallocation از طریق جابجایی می کنند که می تواند

delete پوینتر خود را و در واقع پوینتر را destroy می کند.

در صورتی که dangling pointer جابجایی می کند یعنی مثل پوینتری که در

آدرس که allocate شده بود را می بیند و بعد SAHEL deallocate

می کنیم و می چیدانیم آن آدرس که امکان دارد بعد از این یک

مقدار دیگر `malloc` یا `new` بکنیم و این را می بینیم !!

dangling pointer → deleted obj

pointer → (obj)

pointer → (obj)

برای انتقال یک آرایه `return` آن در `function` می باید یک حقیقت

از سمت `Heap` می آید چون اگر از `stack` استفاده کنیم بعد از

پایان آن `function` همه اطلاعات آن می چیدانیم و آرایه حذف می شود

و برای `function` های دیگر استفاده می شود و از هم اینها جدا است !!

مثال: `int* Addone (int A[] (int n)`

```
{
    int* ptr;
    ptr = new int[n];
    for (int i=0; i<n; i++)
        *(ptr+i) = A[i] + 1;
    return ptr;
}
```

SAHEL

```
int main ()
{
```

Subject

Year Month Day

```
int A[10];
```

```
for (int i = 0; i < 10; i++)
```

```
    A[i] = i + 1;
```

```
int * pointer;
```

```
pointer = addOne(A);
```

```
for (int j = 0; j < 10; j++)
```

```
    cout << *(pointer + j) << " ";
```