- وقتی قبل از متغیری علامت & قرار می دهیم بدین معناست که می خواهیم آدرس آن متغیر را در حافظه بخوانیم. در تابع max1 نیز در آرگومان های تابع به جای گرفتن خود متغیر ابتدا محل قرارگیری متغیرها را در حافظه پیدا می کنیم سپس عملیات مورد نظر را بر روی مقادیری که در آن خانه هستند انجام می دهیم؛ مقادیر بازگشتی در تابع max1 نیز نشان می دهد که دیگر نیازی به استفاده از & نداریم زیرا کاری به آدرس متغیرها نداریم بلکه با خود مقادیر کار داریم البته شایان ذکر است که تابع max1 بدون گذاشتن & در آرگومان ها به درستی کار می کند در واقع به طور مستقیم با مقادیر کار می کند همانطور که در تابع main نوشته شده است.

- وقتی قبل از متغیری علامت * قرار می دهیم در واقع به آن متغیر اجازه می دهیم که یک آدرسی از حافظه را در خودش ذخیره کند پس عملا به صورت منطقی می توان گفت که در صورت نیاز &b و *a که در واقع با این کار آدرس متغیر b را در a ذخیره کردیم که یک pointer است و به آدرس ها اشاره می کند پس در تابع max2 هم می توان گفت که آرگومان های تابع آدرس یک متغیر را می گیرد و پس در مقایسه مقادیر داخل خانه ها را با هم مقایسه می کند اما برخلاف تابع 1 اگر در گذاشتن pointer در آرگومان های تابع دقت نکنیم با ارور مواجه می شویم

- تفاوت های تابع max1 با خود متغیرها کار می کند ولی تابع max2 آدرس متغیرها را می گیرد پس با مقادیر داخل آن ها کار می کند ولی در نهایت جواب هردو یکسان است.

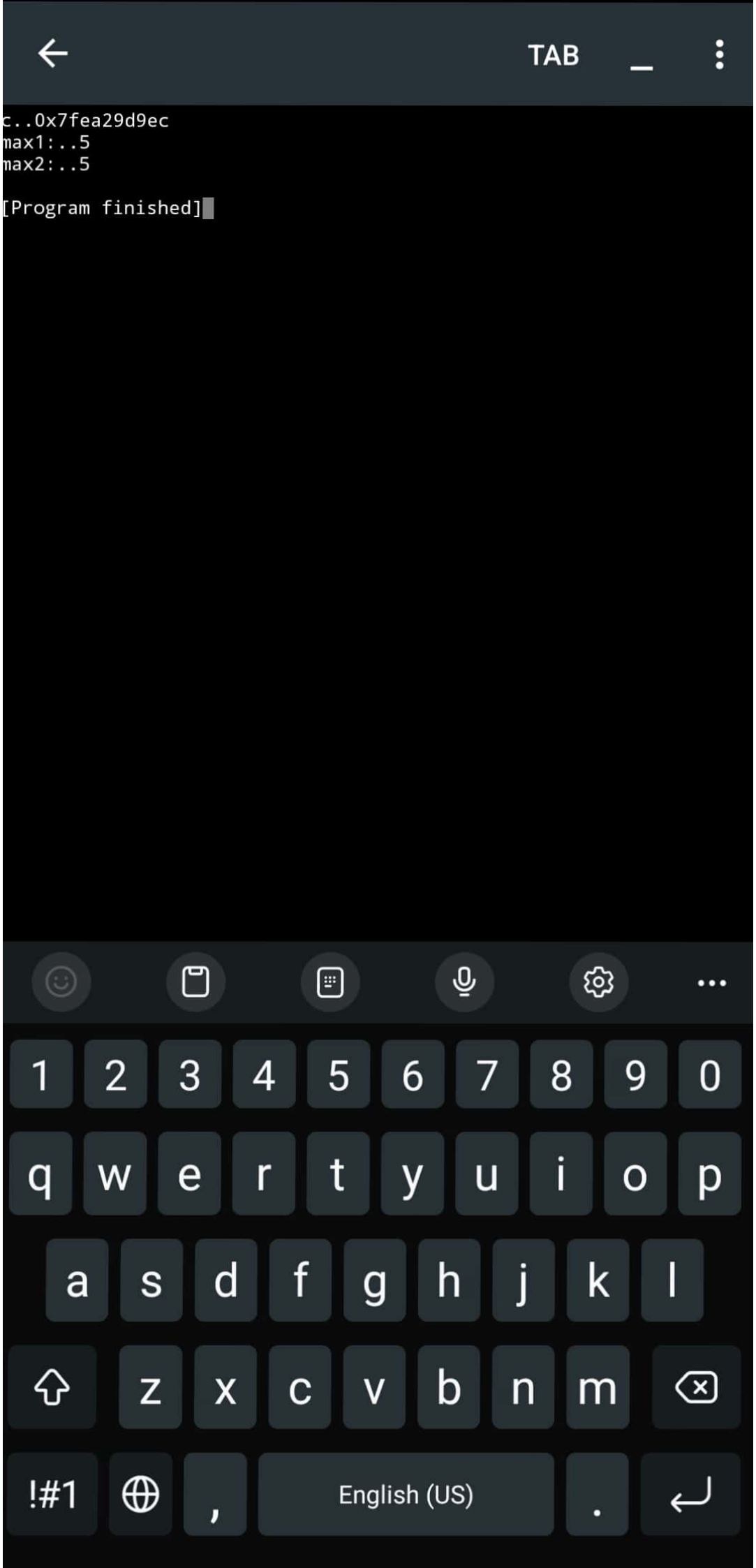
new*

```cpp
#include <iostream>
 using namespace std;
 int &max1(int &m ,int &n) {return (m > n? m:n);}
  int &max2(int *m ,int *n) {return (*m > *n? *m:*n);
}
  int main(){
      int a=5,b=3;
      int *c=&a;
      cout <<"c.."<< c <<endl;
      cout <<"max1:.."<< max1(a,b) <<endl;
      cout <<"max2:.."<< max2(c,&b)<<endl;

      }

```

Tab | { | } | : | ; | " |


TAB
c..0x7fea29d9ec
max1:..5
max2:..5
[Program finished]
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
!#1
English (US)