

سوال ۲

اشکان سلیمانی ۹۵۱۰۵۶۳۵

قطعه کد ۱:

در ابتدا تمامی خطوطی که malloc یا calloc دارند اما casting نشده اند چرا که خروجی این دو void* است خطای کامپایل میخورند! (خطوط ۱ و ۴ و ۵ و ۷ و ۸ و ۹)

خوب برای هر خط پنج خروجی داریم میدانیم که خروجی دوم همگی نشان دهنده اندازه مقدار پوینتر است که بسته به سیستم یا ۴ یا ۸ byte می باشد و برای همه خط ها مشترک و برابر!

خروجی سوم و چهارم هر خط هم که آدرس حافظه allocate و خونه ی بعدی اونه!

و خروجی پنجم هر خط مقدار موجود در خانه اول مجموعه خانه های allocate شده است که calloc باشد مطمئنا در ابتدا مقدار صفر را به خود میگیرد! و اما malloc نا مشخص است!

حال کافی است که خروجی اول که اندازه ی خونه ی اول allocate شده است و مقدار طول هر حافظه انتخاب شده را بررسی کنیم!

۱: با فرض عدم خطای کامپایل (به طور مشابه برای بقیه خطوط) :

خروجی اول: ۴ (به اندازه int) - طول حافظه: $۸ * ۱ = ۸$ (size of char = 1)

۲: مشابه خط اول

۳: خروجی اول: ۴ - طول حافظه: $۸ * ۴ = ۳۲$

۴: خروجی اول: ۱ (به اندازه char) - طول حافظه: $۸ * ۱ = ۸$ (size of char = 1)

۵: خروجی اول: ۱ - طول حافظه: ۳۲

۶: خطای کامپایل casting نادرست

۷: خروجی اول: ۴ (به اندازه float) - طول حافظه: $۸ * ۴ = ۳۲$ (size of int = 4)

۸: خروجی اول: ۸ (به اندازه double) - طول حافظه: $۸ * ۱ = ۸$

۹: خروجی اول: ۴ (به اندازه int) - طول حافظه: $۸ * ۴ = ۳۲$

قطعه کد ۲:

Runtime error می شود چرا که پوینتر c به b اشاره میکند که b خود به int است و نمی تواند خانه آن free شود! پس عبارت free(c) مسبب خطای runtime می شود!

قطعه کد ۳:

می دانیم که پوینتر به خانه اول مجموعه حافظه allocate شده اشاره می کند! و مقدار آن برابر با آدرس خانه ی اول است. حال در printf :

%d : مقدار آدرس را چاپ میکند.

%u : مقدار آدرس را با علامت مثبت چاپ می کند!

%p : مقدار آدرس را برای یک پوینتر در مبنای ۱۶ چاپ میکند!

که در نهایت آدرس خانه اول به طرق مختلف چاپ می شود.

قطعه کد ۴:

اول از همه باید در خط تعریف realloc b cast بشود به char*

خروجی:

DE13C0.....

DE13C4.....

DE13C0.....

DE13C1.....

صرفاً چهار آدرس چاپ می شود بدین قسم که آدرس اول و سوم یکی می باشد چرا که اندازه مقدار alloc شده در بار اول بیشتر از مقدار realloc شده است پس در realloc همان خانه های قبلی منتها خانه های کم تری alloc می شود. از طرفی میدانیم که در خط دوم و چهارم به ترتیب آدرس خانه های بعدی به ترتیب خط اول و سوم چاپ می شود. چون که پوینتر a از جنس *int است فاصله ی بین دو خانه متوالی آن چهار واحد و چون b از نوع *char است فاصله ی بین دو خانه متوالی آن یک واحد است پس خروجی های خط دوم چهارم هر یک به ترتیب باید ۴ واحد و ۱ واحد بزرگتر از آدرس خانه اول باشند.