



printf

քանի որ putnbr-ը և putstr-ը բավարար
չեն

*Հակիրճ. այս նախագիծը բավականին պարզ է: Պետք է
վերակոդավորել printf-ը: Հուսով ենք՝ ապագայում կկարողանաք այն
օգտագործել՝ առանց վախենալու, որ դա կհամարվի խարդախություն:
Հիմնականում կսովորեք՝ ինչպես օգտագործել վարիադիկ
արգումենտներ:*

Ցանկ

I	Նախաբան	2
II	Ընդհանուր ցուցումներ	3
III	Պարտադիր մաս	5
IV	Բոնուսային մաս	6

Գլուխ I

Նախաբան

C-ում `printf` ֆունկցիայի բազմակողմանիությունը ծրագրավորման փորձի մեծ հնարավորություն է ընձեռում: Սա միջին բարդության նախագիծ է, որը թույլ կտա բացահայտել C-ում վարիադիկ ֆունկցիաները:

Հաջող `ft_printf`-ի համար ամենակարևորն է ունենալ լավ կառուցված և ընդարձակելի կոդ:

Գլուխ II

Ընդհանուր ցուցումներ

- Ձեր նախագիծը պետք է գրվի Norm-ին համապատասխան: Եթե ունեք բոնուսային ֆայլեր/ֆունկցիաներ, դրանք ևս ստուգվելու են Norm-ով, և Norm-ի սխալի դեպքում կստանաք 0 միավոր:
- Բացառությամբ չսահմանված վարքագծերի (*սեզմենտացիայի սխալ, bus սխալ, կրկնակի free* և այլն)՝ ձեր ֆունկցիաները չպետք է անսպասելի ավարտվեն: Եթե դա տեղի ունենա, նախագիծը կհամարվի ոչ ֆունկցիոնալ և գնահատման ժամանակ կստանաք 0 միավոր:
- Հատկացված դինամիկ բաշխվող հիշողությունն անհրաժեշտության դեպքում պետք է ամբողջությամբ պատշաճ կերպով ազատվի: Արտահոսքերն անընդունելի են:
- Եթե նյութը պահանջում է, ապա պետք է հանձնել Makefile, որը ձեր ելակետային ֆայլը կկազմարկի պահանջվող ելքի՝ -Wall, -Wextra և -Werror դրոշակներով, իսկ ձեր Makefile-ը չպետք է վերակապի:
- Ձեր Makefile-ն պետք է պարունակի առնվազն \$(NAME), all, clean, fclean և re կանոնները:
- Ձեր նախագծի հետ բոնուսներ հանձնելու համար պետք է Makefile-ում ներառել bonus կանոն, որը կավելացնի բոլոր հնարավոր վերնագրերը, գրադարանները կամ ֆունկցիաները, որոնք արգելված են նախագծի հիմնական մասում: Բոնուսները պետք է լինեն առանձին ֆայլում՝ _bonus.{c/h}-ում: Պարտադիր և բոնուսային մասերի գնահատումը կատարվում է իրարից անկախ:
- Եթե նախագիծը թույլ է տալիս օգտագործել ձեր libft-ը, ապա պետք է պատճենեք դրա ելակետային ֆայլերը և համապատասխան Makefile-ը libft պանակում՝ իր համապատասխան Makefile-ով: Ձեր նախագծի Makefile-ը պետք է կազմարկի գրադարանը՝ օգտագործելով դրա Makefile-ը, իսկ հետո՝ կազմարկի նախագիծը:
- Խրախուսելի է ձեր նախագծի համար թեստային ծրագրերի ստեղծումը, որոնք, սակայն, **հաշվի չեն առնվի և չեն գնահատվի**: Դա հնարավորություն կընձեռնի հեշտությամբ թեստավորել ձեր և ձեր ընկերների աշխատանքը: Այս թեստերը

հատկապես օգտակար կլինեն պաշտպանության ժամանակ: Ավելին, պաշտպանության ընթացքում ազատորեն կարող եք օգտագործել ինչպես ձեր, այնպես էլ այն ուսանողների թեստերը, ում աշխատանքը գնահատում եք:

- Ձեր աշխատանքը պետք է ներառել հանձնարարված git պահոցում: Կգնահատվի միայն git պահոցի աշխատանքը: Եթե նախատեսվում է, որ Deepthought-ն է գնահատելու ձեր աշխատանքը, ապա դա կարվի ձեր ընկերների գնահատումից հետո: Եթե Deepthought-ի գնահատման ընթացքում աշխատանքի ցանկացած մասում որևէ սխալ ի հայտ գա, գնահատումը կընդհատվի:

Գլուխ III

Պարտադիր մաս

Ծրագրի անուն	libftprintf.a
Հանձնվելիք ֆայլեր	*.c, */*.c, *.h, */*.h, Makefile
Ստեղծելի ֆայլ	all, clean, fclean, re, bonus
Արտաքին ֆունկցիաներ	malloc, free, write, va_start, va_arg, va_copy, va_end
Libft թույլատրված	yes
Նկարագրություն	Գրել ft_printf ֆունկցիա պարունակող գրադարան, որը կրկնօրինակում է իրական printf-ը:

- ft_printf-ի նախատիպը պետք է լինի `int int ft_printf(const char *, ...);`
- Դուք պետք է վերակոդավորեք libc-ի printf ֆունկցիան:
- Այն չպետք է կատարի բուֆերային կառավարում, ինչպես իրական printf-ը:
- Այն պետք է կատարի հետևյալ փոխարկումները՝ `cspdiuxX%`:
- Այն պետք է կատարի հետևյալ դրոշակների կամայական համադրություն՝ `'-0.*'` և դաշտի նվազագույն լայնություն՝ կամայական փոխարկումներով:
- Այն կհամեմատվի իրական printf-ի հետ:
- Դուք պետք է օգտագործեք `ar` հրամանը ձեր գրադարանը ստեղծելու համար, `libtool` հրամանի օգտագործումն արգելված է:



man 3 printf / man 3 stdarg

Գլուխ IV

Բոնուսային մաս

- Եթե պարտադիր մասը կատարյալ չէ, նույնիսկ մի՛ մտածեք բոնուսի մասին:
- Պարտադիր չէ անել բոլոր բոնուսները:
- Կատարե՛ք հետևյալ փոխարկումներից մեկը կամ մի քանիսը՝ *nfge*:
- Կատարե՛ք հետևյալ դրոշակներից մեկը կամ մի քանիսը՝ 1 11 h hh:
- Կատարե՛ք բոլոր հետևյալ դրոշակները՝ ‘# +’ (այո՛, դրանցից մեկը բացատ է):