

</> 1	CHELLO	Hello World	LẬP TRÌNH C++ CƠ BẢN
</> 2	CPP0101	TÍNH TỔNG 1 ĐẾN N	LẬP TRÌNH C++ CƠ BẢN
</> 3	CPP0102	CHỮ HOA – CHỮ THƯỜNG	LẬP TRÌNH C++ CƠ BẢN
</> 4	CPP0103	TÍNH TỔNG PHÂN THỨC	LẬP TRÌNH C++ CƠ BẢN
</> 5	CPP0104	TÍNH TỔNG GIAI THỪA	LẬP TRÌNH C++ CƠ BẢN
</> 6	CPP0105	SỐ LỘC PHÁT	LẬP TRÌNH C++ CƠ BẢN
</> 7	CPP0106	SỐ THUẬN NGHỊCH	LẬP TRÌNH C++ CƠ BẢN
</> 8	CPP0107	CHẤM ĐIỂM TRẮC NGHIỆM	LẬP TRÌNH C++ CƠ BẢN
</> 9	CPP0109	CÂN BẰNG CHẴN LẺ	LẬP TRÌNH C++ CƠ BẢN
</> 10	CPP0110	MÃ SỐ QUỐC GIA	LẬP TRÌNH C++ CƠ BẢN
</> 11	CPP0111	SỐ LIÊN KÈ	LẬP TRÌNH C++ CƠ BẢN
</> 12	CPP0112	KHOẢNG CÁCH	LẬP TRÌNH C++ CƠ BẢN
</> 13	CPP0113	SỐ MAY MẮN	LẬP TRÌNH C++ CƠ BẢN
</> 14	CPP0115	PHÂN TÍCH THỪA SỐ NGUYÊN TỐ - 1	LẬP TRÌNH C++ CƠ BẢN
</> 15	CPP0117	TÍNH TỔNG CHỮ SỐ	LẬP TRÌNH C++ CƠ BẢN
</> 16	CPP0121	ƯỚC SỐ CHUNG - BỘI SỐ CHUNG	LẬP TRÌNH C++ CƠ BẢN
</> 17	CPP0122	ƯỚC SỐ CHUNG LỚN NHẤT CỦA N SỐ NGUYÊN DƯƠNG ĐẦU TIÊN	LẬP TRÌNH C++ CƠ BẢN

</>18	CPP0123	KIỂM TRA NGUYÊN TỐ	LẬP TRÌNH C++ C	
</>	19	CPP0124	PHÂN TÍCH THỪA SỐ NGUYÊN TỐ - 2	LẬP TRÌNH C++ C
</>	20	CPP0125	LIỆT KÊ SỐ NGUYÊN TỐ - 1	LẬP TRÌNH C++ C
</>	21	CPP0126	LIỆT KÊ SỐ NGUYÊN TỐ - 2	LẬP TRÌNH C++ C
</>	22	CPP0127	CẶP SỐ NGUYÊN TỐ ĐẦU TIÊN CÓ TỔNG BẰNG N	LẬP TRÌNH C++ C
</>	23	CPP0130	ƯỚC SỐ NGUYÊN TỐ	LẬP TRÌNH C++ C
</>	24	CPP0132	ƯỚC SỐ NGUYÊN TỐ LỚN NHẤT	LẬP TRÌNH C++ C
</>	25	CPP0133	ƯỚC SỐ NGUYÊN TỐ NHỎ HƠN N	LẬP TRÌNH C++ C
</>	26	CPP0134	ƯỚC SỐ NGUYÊN TỐ THỨ K	LẬP TRÌNH C++ C
</>	27	CPP0135	LIỆT KÊ SỐ CÓ BA ƯỚC SỐ	LẬP TRÌNH C++ C
</>	28	CPP0138	CẶP SỐ NGUYÊN TỐ	LẬP TRÌNH C++ C
</>	29	CPP0143	SỐ FIBONACCI THỨ N	LẬP TRÌNH C++ C
</>	30	CPP0152	CHIA DƯ	LẬP TRÌNH C++ C
</>	31	CPP0153	CHIA DƯ TỪ 1 ĐẾN N	LẬP TRÌNH C++ C
</>	32	CPP0154	TỔNG CHIA DƯ CHO K	LẬP TRÌNH C++ C
</>	33	CPP0201	CHÊNH LỆCH NHỎ NHẤT	MẢNG TRỖ
</>	34	CPP0202	KHOẢNG CÁCH NHỎ NHẤT	MẢNG TRỖ
</>	35	CPP0203	SỐ NHỎ NHẤT CHƯA XUẤT HIỆN	MẢNG TRỖ

</>	36	CPP0205	DÃY TAM GIÁC	MÃNG TRỎ
</>	37	CPP0206	PHẦN TỬ LỚN NHẤT	MÃNG TRỎ
</>	38	CPP0207	QUAY VÒNG DÃY SỐ 1	MÃNG TRỎ
</>	39	CPP0208	PHẦN TỬ NHỎ NHẤT THỨ K	MÃNG TRỎ
</>	40	CPP0209	TÍNH TỔNG TRONG KHOẢNG	MÃNG TRỎ
</>	41	CPP0219	BIẾN ĐỔI NHỊ PHÂN	MÃNG TRỎ
</>	42	CPP0220	BIÊN CỬA MA TRẬN	MÃNG TRỎ
</>	43	CPP0227	IN MA TRẬN - 1	MÃNG TRỎ
</>	44	CPP0230	MA TRẬN NHỊ PHÂN	MÃNG TRỎ
</>	45	CPP0244	LIỆT KÊ SỐ KHÁC NHAU	MÃNG TRỎ
</>	46	CPP0259	TÍCH MA TRẬN	MÃNG TRỎ
</>	47	CPP0274	ĐẾM SỐ PHẦN TỬ LẶP LẠI	MÃNG TRỎ
</>	48	CPP0308	KÝ TỰ KHÔNG LẶP	XỬ LÝ KÝ TỰ
</>	49	CPP0309	ĐẾM TỪ	XỬ LÝ KÝ TỰ
</>	50	CPP0312	XÂU PANGRAM	XỬ LÝ KÝ TỰ
</>	51	CPP0313	LOẠI BỎ TỪ TRONG XÂU	XỬ LÝ KÝ TỰ
</>	52	CPP0314	CHÚC MỪNG NĂM MỚI	XỬ LÝ KÝ TỰ
</>	53	CPP0317	SỐ ĐẸP	XỬ LÝ KÝ TỰ

</>	54	CPP0319	NHỎ NHẤT - LỚN NHẤT	XỬ LÝ KÝ TỰ
</>	55	CPP0325	CHIA HẾT CHO 11	XỬ LÝ KÝ TỰ
</>	56	CPP0327	CHIA HẾT CHO 5	XỬ LÝ KÝ TỰ
</>	57	CPP0332	ĐỊA CHỈ EMAIL - 1	XỬ LÝ KÝ TỰ
</>	58	CPP0333	CHUẨN HÓA HỌ TÊN	XỬ LÝ KÝ TỰ
</>	59	CPP0339	ĐẦU CUỐI GIỐNG NHAU	XỬ LÝ KÝ TỰ
</>	60	CPP0343	DẪY ƯU THỂ	XỬ LÝ KÝ TỰ
</>	61	CPP0354	MÃ HÓA	XỬ LÝ KÝ TỰ
</>	62	CPP0371	LOẠI BỎ NGUYÊN ÂM	XỬ LÝ KÝ TỰ
</>	63	CPP0412	SẮP XẾP 0 -1 - 2	KỶ TH XẾP V KIỂM
</>	64	CPP0413	SẮP XẾP XEN KẼ - 1	KỶ TH XẾP V KIỂM
</>	65	CPP0414	SẮP XẾP CHỮ SỐ	KỶ TH XẾP V KIỂM
</>	66	CPP0415	TÍNH TÍCH	KỶ TH XẾP V KIỂM
</>	67	CPP0416	ĐẾM CẶP PHẦN TỬ CÓ TỔNG BẰNG K	KỶ TH XẾP V KIỂM
</>	68	CPP0421	SẮP ĐẶT DẪY SỐ	KỶ TH XẾP V KIỂM
</>	69	CPP0422	SỐ 0 Ở CUỐI DẪY	KỶ TH XẾP V KIỂM

</>	70	CPP0424	GHÉP DẪY SỐ	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	71	CPP0428	TRỘN HAI DẪY VÀ SẮP XẾP	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	72	CPP0430	BỔ SUNG PHẦN TỬ	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	73	CPP0441	TÌM KIẾM TUẦN TỰ	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	74	CPP0442	TÌM KIẾM NHỊ PHÂN	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	75	CPP0443	SỐ NHỎ NHẤT CÒN THIẾU	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	76	CPP0444	TÌM KIẾM TRONG DẪY SẮP XẾP VÒNG	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	77	CPP0445	SỐ NHỎ NHẤT VÀ NHỎ THỨ HAI	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	78	CPP0446	TỔNG GẦN 0 NHẤT	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	79	CPP0447	LIỆT KÊ K PHẦN TỬ LỚN NHẤT	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	80	CPP0448	ĐẾM SỐ LẦN XUẤT HIỆN	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	81	CPP0450	PHẦN TỬ ĐẦU TIÊN LẶP LẠI	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	82	CPP0501	CẤU TRÚC ĐIỂM	CẤU TRÚC MẢNG TRÚC
</>	83	CPP0502	CẤU TRÚC THÍ SINH	CẤU TRÚC MẢNG TRÚC

</>	84	CPP0503	CẤU TRÚC PHÂN SỐ	CẤU TRÚC MẢNG TRƯỚC
</>	85	CPP0504	CẤU TRÚC SINH VIÊN	CẤU TRÚC MẢNG TRƯỚC
</>	86	CPP0505	CẤU TRÚC NHÂN VIÊN	CẤU TRÚC MẢNG TRƯỚC
</>	87	CPP0507	TÍNH TỔNG HAI PHÂN SỐ	CẤU TRÚC MẢNG TRƯỚC
</>	88	CPP0517	DANH SÁCH NHÂN VIÊN	CẤU TRÚC MẢNG TRƯỚC
</>	89	CPP0601	KHAI BÁO LỚP SINH VIÊN - 1	LỚP V TUỖN
</>	90	CPP0602	KHAI BÁO LỚP SINH VIÊN - 2	LỚP V TUỖN
</>	91	CPP0603	KHAI BÁO LỚP SINH VIÊN - 3	LỚP V TUỖN
</>	92	CPP0605	KHAI BÁO LỚP PHÂN SỐ	LỚP V TUỖN
</>	93	CPP0606	KHAI BÁO LỚP NHÂN VIÊN	LỚP V TUỖN
</>	94	CPP0610	TÍNH TỔNG HAI ĐỐI TƯỢNG PHÂN SỐ	LỚP V TUỖN
</>	95	CPP0711	LIỆT KÊ XÂU NHỊ PHÂN	CÁC NÂNG
</>	96	CPP0714	HOÁN VỊ LIÊN KÈ PHÍA TRƯỚC	CÁC NÂNG
</>	97	CPP0723	TÍCH GIAI THỪA CÁC CHỮ SỐ	CÁC NÂNG
</>	98	CPP0741	TÍNH LŨY THỪA	CÁC NÂNG
</>	99	CPP0743	ĐẢO TỪ	CÁC NÂNG
</>	100	CPP0801	SAO CHÉP TỆP TIN	LẬP T C++ C

</>	101	<a href="#">CPP204C4000</a>	<a href="#">CẤU TRÚC SINH VIÊN</a>	CẤU TRÚC MẢNG TRÚC
</>	103	<a href="#">CPP0108</a>	<a href="#">SỐ TĂNG GIẢM</a>	LẬP TRÌNH C++ C
</>	104	<a href="#">CPP0116</a>	<a href="#">ƯỚC SỐ NGUYÊN TỐ NHỎ NHẤT</a>	LẬP TRÌNH C++ C
</>	105	<a href="#">CPP0118</a>	<a href="#">SỐ SPHENIC</a>	LẬP TRÌNH C++ C
</>	106	<a href="#">CPP0119</a>	<a href="#">ƯỚC SỐ CHIA HẾT CHO 2</a>	LẬP TRÌNH C++ C
</>	107	<a href="#">CPP0120</a>	<a href="#">CHIA HẾT CHO A VÀ B</a>	LẬP TRÌNH C++ C
</>	108	<a href="#">CPP0131</a>	<a href="#">ƯỚC SỐ NGUYÊN TỐ NHỎ NHẤT</a>	LẬP TRÌNH C++ C
</>	109	<a href="#">CPP0136</a>	<a href="#">ĐẾM SỐ CÓ BA ƯỚC SỐ</a>	LẬP TRÌNH C++ C
</>	110	<a href="#">CPP0141</a>	<a href="#">KIỂM TRA SỐ FIBONACCI</a>	LẬP TRÌNH C++ C
</>	111	<a href="#">CPP0142</a>	<a href="#">NGUYÊN TỐ CÙNG NHAU</a>	LẬP TRÌNH C++ C
</>	112	<a href="#">CPP0173</a>	<a href="#">CHIA HẾT CHO BA SỐ NGUYÊN</a>	LẬP TRÌNH C++ C
</>	113	<a href="#">CPP0181</a>	<a href="#">SỐ LẶP LẠI</a>	LẬP TRÌNH C++ C
</>	114	<a href="#">CPP0204</a>	<a href="#">ĐẾM SỐ NGUYÊN TỐ</a>	MẢNG TRỞ
</>	115	<a href="#">CPP0210</a>	<a href="#">HIỆU LỚN NHẤT CỦA CẤP PHẦN TỬ ĐÚNG THỨ TỰ</a>	MẢNG TRỞ
</>	116	<a href="#">CPP0212</a>	<a href="#">TÍNH GIÁ TRỊ ĐA THỨC</a>	MẢNG TRỞ
</>	117	<a href="#">CPP0213</a>	<a href="#">KIỂM TRA DÃY FIBONACCI</a>	LẬP TRÌNH C++ C
</>	118	<a href="#">CPP0216</a>	<a href="#">DÃY MOUNTAIN</a>	MẢNG TRỞ

</>	119	<a href="#">CPP0217</a>	<a href="#">PHẦN TỬ NHỎ NHẤT THỨ K CỦA MA TRẬN</a>	MẢNG TRỎ
</>	120	<a href="#">CPP0221</a>	<a href="#">QUAY MA TRẬN</a>	MẢNG TRỎ
</>	121	<a href="#">CPP0222</a>	<a href="#">ĐẾM PHẦN TỬ GIỐNG NHAU</a>	MẢNG TRỎ
</>	122	<a href="#">CPP0223</a>	<a href="#">MA TRẬN XOẮN ỐC - 1</a>	MẢNG TRỎ
</>	124	<a href="#">CPP0225</a>	<a href="#">BIẾN ĐỔI MA TRẬN</a>	MẢNG TRỎ
</>	126	<a href="#">CPP0241</a>	<a href="#">BIẾN ĐỔI DÃY SỐ</a>	MẢNG TRỎ
</>	127	<a href="#">CPP0242</a>	<a href="#">DÃY SỐ NHỊ PHÂN</a>	MẢNG TRỎ
</>	128	<a href="#">CPP0260</a>	<a href="#">MA TRẬN XOẮN ỐC - 4</a>	MẢNG TRỎ
</>	129	<a href="#">CPP0261</a>	<a href="#">CỬA SỔ TRƯỢT</a>	MẢNG TRỎ
</>	130	<a href="#">CPP0273</a>	<a href="#">ĐIỂM CÂN BẰNG</a>	MẢNG TRỎ
</>	131	<a href="#">CPP0307</a>	<a href="#">HIỆU CỦA HAI TẬP TỪ</a>	XỬ LÝ KÝ TỰ
</>	132	<a href="#">CPP0310</a>	<a href="#">TỔNG LỚN NHẤT VÀ NHỎ NHẤT</a>	XỬ LÝ KÝ TỰ
</>	133	<a href="#">CPP0311</a>	<a href="#">SẮP ĐẶT XÂU KÝ TỰ - 1</a>	XỬ LÝ KÝ TỰ
</>	134	<a href="#">CPP0315</a>	<a href="#">ĐỔI CHỖ CHỮ SỐ</a>	XỬ LÝ KÝ TỰ
</>	135	<a href="#">CPP0316</a>	<a href="#">SỐ MAY MẮN - 2</a>	XỬ LÝ KÝ TỰ
</>	136	<a href="#">CPP0318</a>	<a href="#">BIẾN SỐ ĐẸP</a>	XỬ LÝ KÝ TỰ
</>	137	<a href="#">CPP0320</a>	<a href="#">SỐ ĐẦY ĐỦ</a>	XỬ LÝ KÝ TỰ
</>	138	<a href="#">CPP0321</a>	<a href="#">HIỆU HAI SỐ NGUYÊN LỚN</a>	XỬ LÝ KÝ TỰ



</>	139	<a href="#">CPP0322</a>	<a href="#">TỔNG HAI SỐ NGUYÊN LỚN</a>	XỬ LÝ KÝ TỰ
</>	140	<a href="#">CPP0323</a>	<a href="#">PHÉP CHIA DƯ CỦA SỐ NGUYÊN LỚN</a>	XỬ LÝ KÝ TỰ
</>	141	<a href="#">CPP0324</a>	<a href="#">PHÉP CHIA DƯ CỦA LŨY THỪA SỐ NGUYÊN LỚN</a>	XỬ LÝ KÝ TỰ
</>	142	<a href="#">CPP0334</a>	<a href="#">TÍNH TỔNG CÁC SỐ TRONG XÂU</a>	XỬ LÝ KÝ TỰ
</>	143	<a href="#">CPP0335</a>	<a href="#">TÌM SỐ LỚN NHẤT TRONG XÂU</a>	XỬ LÝ KÝ TỰ
</>	144	<a href="#">CPP0338</a>	<a href="#">ĐẾM XÂU CON</a>	XỬ LÝ KÝ TỰ
</>	145	<a href="#">CPP0342</a>	<a href="#">TÁCH CHỮ SỐ</a>	XỬ LÝ KÝ TỰ
</>	146	<a href="#">CPP0351</a>	<a href="#">CHUẨN HÓA HỌ TÊN</a>	XỬ LÝ KÝ TỰ
</>	147	<a href="#">CPP0352</a>	<a href="#">ĐỊA CHỈ EMAIL - 2</a>	XỬ LÝ KÝ TỰ
</>	148	<a href="#">CPP0353</a>	<a href="#">ĐIỆN THOẠI CỤC GẠCH</a>	XỬ LÝ KÝ TỰ
</>	149	<a href="#">CPP0355</a>	<a href="#">XỬ LÝ VĂN BẢN</a>	XỬ LÝ KÝ TỰ
</>	150	<a href="#">CPP0417</a>	<a href="#">SẮP XẾP LẠI DÃY CON</a>	KỶ TH XẾP V KIỂM
</>	151	<a href="#">CPP0418</a>	<a href="#">HỢP VÀ GIAO CỦA HAI DÃY SỐ -1</a>	KỶ TH XẾP V KIỂM
</>	152	<a href="#">CPP0419</a>	<a href="#">HỢP VÀ GIAO CỦA HAI DÃY SỐ - 2</a>	KỶ TH XẾP V KIỂM
</>	153	<a href="#">CPP0420</a>	<a href="#">SẮP XẾP THEO KHOẢNG CÁCH</a>	KỶ TH XẾP V KIỂM
</>	154	<a href="#">CPP0423</a>	<a href="#">SỐ NHỎ HƠN K</a>	KỶ TH XẾP V KIỂM

</>	155	<a href="#">CPP0425</a>	<a href="#">SẮP XẾP CHẴN LẺ</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	156	<a href="#">CPP0426</a>	<a href="#">SẮP XẾP XEN KẼ - 2</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	157	<a href="#">CPP0427</a>	<a href="#">NHÂN ĐÔI CẶP SỐ BẰNG NHAU</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	158	<a href="#">CPP0429</a>	<a href="#">SỬA ĐÈN</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	159	<a href="#">CPP0431</a>	<a href="#">ĐẾM CẶP PHẦN TỬ CÓ HIỆU NHỎ HƠN K</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	160	<a href="#">CPP0432</a>	<a href="#">SẮP XẾP THEO CHỮ SỐ</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	161	<a href="#">CPP0434</a>	<a href="#">BIẾN ĐỔI DÃY SỐ - 2</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	162	<a href="#">CPP0449</a>	<a href="#">KHOẢNG CÁCH BẰNG X</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	163	<a href="#">CPP0451</a>	<a href="#">PHẦN TỬ GẦN NHẤT</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	164	<a href="#">CPP0453</a>	<a href="#">NHỎ NHẤT THỨ K</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	165	<a href="#">CPP0454</a>	<a href="#">TAM GIÁC VUÔNG</a>	KỸ THUẬT XẾP VÀ KIỂM TRA
</>	166	<a href="#">CPP0512</a>	<a href="#">PHÉP TOÁN VỚI PHẦN SỐ</a>	CẤU TRÚC MẢNG VÀ TRÚC
</>	167	<a href="#">CPP0513</a>	<a href="#">DANH SÁCH SINH VIÊN - 1</a>	CẤU TRÚC MẢNG VÀ TRÚC
</>	168	<a href="#">CPP0514</a>	<a href="#">DANH SÁCH SINH VIÊN - 2</a>	CẤU TRÚC MẢNG VÀ TRÚC

</>	169	<a href="#">CPP0515</a>	<a href="#">DANH SÁCH SINH VIÊN - 3</a>	CẦU 7 MẢNG TRÚC
</>	170	<a href="#">CPP0516</a>	<a href="#">SẮP XẾP DANH SÁCH MẶT HÀNG</a>	CẦU 7 MẢNG TRÚC
</>	171	<a href="#">CPP0518</a>	<a href="#">SẮP XẾP DANH SÁCH NHÂN VIÊN</a>	CẦU 7 MẢNG TRÚC
</>	172	<a href="#">CPP0524</a>	<a href="#">BẢNG ĐIỂM THÀNH PHẦN - 1</a>	CẦU 7 MẢNG TRÚC
</>	173	<a href="#">CPP0525</a>	<a href="#">BẢNG ĐIỂM THÀNH PHẦN - 2</a>	CẦU 7 MẢNG TRÚC
</>	174	<a href="#">CPP0526</a>	<a href="#">TRẺ NHẤT – GIÀ NHẤT</a>	CẦU 7 MẢNG TRÚC
</>	175	<a href="#">CPP0527</a>	<a href="#">SẮP XẾP THỜI GIAN</a>	CẦU 7 MẢNG TRÚC
</>	176	<a href="#">CPP0528</a>	<a href="#">DANH SÁCH THỰC TẬP - 1</a>	CẦU 7 MẢNG TRÚC
</>	177	<a href="#">CPP0529</a>	<a href="#">DANH SÁCH THỰC TẬP - 2</a>	CẦU 7 MẢNG TRÚC
</>	178	<a href="#">CPP0534</a>	<a href="#">SỐ THUẬN NGHỊCH GIẢM DẦN</a>	CẦU 7 MẢNG TRÚC
</>	179	<a href="#">CPP0544</a>	<a href="#">DIỆN TÍCH HÌNH TRÒN NGOẠI TIẾP TAM GIÁC</a>	CẦU 7 MẢNG TRÚC
</>	180	<a href="#">CPP0611</a>	<a href="#">DANH SÁCH ĐỐI TƯỢNG SINH VIÊN - 1</a>	LỚP V TUỖN
</>	181	<a href="#">CPP0612</a>	<a href="#">DANH SÁCH ĐỐI TƯỢNG SINH VIÊN - 2</a>	LỚP V TUỖN
</>	182	<a href="#">CPP0613</a>	<a href="#">DANH SÁCH ĐỐI TƯỢNG SINH VIÊN - 3</a>	LỚP V TUỖN
</>	183	<a href="#">CPP0614</a>	<a href="#">DANH SÁCH ĐỐI TƯỢNG NHÂN VIÊN</a>	LỚP V TUỖN

</>	184	<a href="#">CPP0615</a>	<a href="#">SẮP XẾP DANH SÁCH ĐỐI TƯỢNG NHÂN VIÊN</a>	LỚP V TUỖN
</>	185	<a href="#">CPP0620</a>	<a href="#">SẮP XẾP SINH VIÊN THEO LỚP</a>	LỚP V TUỖN
</>	186	<a href="#">CPP0621</a>	<a href="#">SẮP XẾP THEO MÃ SINH VIÊN</a>	LỚP V TUỖN
</>	187	<a href="#">CPP0622</a>	<a href="#">LIỆT KÊ SINH VIÊN THEO LỚP</a>	LỚP V TUỖN
</>	188	<a href="#">CPP0623</a>	<a href="#">LIỆT KÊ SINH VIÊN THEO KHÓA</a>	LỚP V TUỖN
</>	189	<a href="#">CPP0624</a>	<a href="#">LIỆT KÊ SINH VIÊN THEO NGÀNH</a>	LỚP V TUỖN
</>	190	<a href="#">CPP0625</a>	<a href="#">SẮP XẾP DANH SÁCH GIẢNG VIÊN</a>	LỚP V TUỖN
</>	191	<a href="#">CPP0626</a>	<a href="#">DANH SÁCH GIẢNG VIÊN THEO BỘ MÔN</a>	LỚP V TUỖN
</>	192	<a href="#">CPP0627</a>	<a href="#">TÌM KIẾM GIẢNG VIÊN</a>	LỚP V TUỖN
</>	193	<a href="#">CPP0628</a>	<a href="#">DANH SÁCH DOANH NGHIỆP NHÂN SINH VIÊN THỰC TẬP - 1</a>	LỚP V TUỖN
</>	194	<a href="#">CPP0629</a>	<a href="#">DANH SÁCH DOANH NGHIỆP NHÂN SINH VIÊN THỰC TẬP - 2</a>	LỚP V TUỖN
</>	195	<a href="#">CPP0712</a>	<a href="#">LIỆT KÊ TỔ HỢP</a>	CÁC H NÂNG
</>	196	<a href="#">CPP0713</a>	<a href="#">LIỆT KÊ HOÁN VI</a>	CÁC H NÂNG
</>	197	<a href="#">CPP0721</a>	<a href="#">DẪY CON TĂNG DÀI NHẤT</a>	CÁC H NÂNG
</>	198	<a href="#">CPP0722</a>	<a href="#">DẪY SỐ CATALAN</a>	CÁC H NÂNG
</>	199	<a href="#">CPP0725</a>	<a href="#">TÍNH SỐ TỔ HỢP</a>	CÁC H NÂNG
</>	200	<a href="#">CPP0726</a>	<a href="#">PHÉP CHIA DƯ CỦA TÍCH HAI SỐ</a>	CÁC H NÂNG

STT	Mã	Tiêu đề	Nhóm	Chủ đề con	Độ khó
</> 201	CPP0727	TỔNG LỚN NHẤT CỦA DÃY CON KHÔNG KÈ NHAU	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	2
</> 202	CPP0730	TỔNG LỚN NHẤT CỦA DÃY CON LIÊN TỤC	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	2
</> 203	CPP0737	DÃY CON TRUNG BÌNH LỚN NHẤT	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	2
</> 204	CPP0738	CỘNG 1 VÀ NHÂN ĐÔI	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	2
</> 205	CPP0739	GIẢI MÃ TĂNG GIẢM	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng thư viện STL	2
</> 206	CPP0740	TÍCH LỚN NHẤT CỦA DÃY CON LIÊN TỤC	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	2
</> 207	CPP0745	TÍNH SỐ FIBONACCI LỚN	CÁC BÀI TOÁN NÂNG CAO	Kỹ thuật chia và trị	2
</> 208	CPP0747	LOẠI BỎ 100	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	2

STT	Mã	Tiêu đề	Nhóm	Chủ đề con	Độ khó
</> 209	CPP0802	TÍNH TỔNG	CÁC BÀI TOÁN NÂNG CAO	Vào ra trên tệp	2
</> 210	CPP0803	SỐ KHÁC NHAU TRONG FILE	CÁC BÀI TOÁN NÂNG CAO	Vào ra trên tệp	2
</> 211	CPP0804	LIỆT KÊ TỪ KHÁC NHAU	CÁC BÀI TOÁN NÂNG CAO	Vào ra trên tệp	2
</> 212	CPP0805	HỢP VÀ GIAO CỦA HAI FILE VĂN BẢN	CÁC BÀI TOÁN NÂNG CAO	Vào ra trên tệp	2
</> 213	CPP0807	LỚP INTSET	CÁC BÀI TOÁN NÂNG CAO	Vào ra trên tệp	2
</> 214	NNLTC_001	THỐNG KÊ PHẦN TỬ	MẢNG VÀ CON TRỎ	Mảng một chiều	2
</> 215	CPP0129	ƯỚC SỐ CỦA GIAI THỪA	LẬP TRÌNH C++ CƠ BẢN	Ước số và Ước số chung lớn nhất	3
</> 216	CPP0137	LIỆT KÊ SỐ CÓ BA ƯỚC SỐ TRONG KHOẢNG	LẬP TRÌNH C++ CƠ BẢN	Số nguyên tố và áp dụng	3
</> 217	CPP0139	SỐ SMITH	LẬP TRÌNH C++ CƠ BẢN	Số nguyên tố và áp dụng	3

STT	Mã	Tiêu đề	Nhóm	Chủ đề con	Độ khó
</> 218	CPP0156	ĐẾM SỐ NGHIỆM CỦA PHƯƠNG TRÌNH ĐỒNG DƯ	LẬP TRÌNH C++ CƠ BẢN	Phép chia dư và áp dụng	3
</> 219	CPP0211	KHOẢNG CÁCH XA NHẤT	MẢNG VÀ CON TRỎ	Mảng một chiều	3
</> 220	CPP0218	ĐỒNG DƯ VỚI K	MẢNG VÀ CON TRỎ	Mảng một chiều	3
</> 224	CPP0243	SẮP ĐẶT HAI DÂY SỐ	MẢNG VÀ CON TRỎ	Mảng một chiều	3
</> 225	CPP0262	TẬP HỢP NGUYÊN TỐ CÙNG NHAU	MẢNG VÀ CON TRỎ	Mảng một chiều	3
</> 226	CPP0270	TÍCH CHẬP	MẢNG VÀ CON TRỎ	Mảng hai chiều	3
</> 228	CPP0272	TÍNH TOÁN TRÊN DÂY SỐ	MẢNG VÀ CON TRỎ	Mảng một chiều	3
</> 229	CPP0326	TÍCH HAI SỐ NGUYÊN LỚN	XỬ LÝ XÂU KÝ TỰ	Xử lý số nguyên lớn	3
</> 230	CPP0328	CHIA HẾT CHO 8	XỬ LÝ XÂU KÝ TỰ	Xử lý số nguyên lớn	3
</> 234	CPP0433	SẮP XẾP THEO SỐ LẦN XUẤT HIỆN	KỸ THUẬT SẮP XẾP	Ứng dụng sắp xếp	3

STT	Mã	Tiêu đề	Nhóm	Chủ đề con	Độ khó
			VÀ TÌM KIẾM		
</> 235	CPP0436	SỐ NHỎ NHẤT LỚN HƠN $A[i]$	KỸ THUẬT SẮP XẾP VÀ TÌM KIẾM	Ứng dụng sắp xếp	3
</> 236	CPP0452	PHẦN TỬ CHUNG CỦA BA DÂY SỐ	KỸ THUẬT SẮP XẾP VÀ TÌM KIẾM	Ứng dụng tìm kiếm	3
</> 237	CPP0531	BỐN ĐIỂM TRÊN MẶT PHẪNG	CẤU TRÚC VÀ MẢNG CẤU TRÚC	Ứng dụng cấu trúc	3
</> 238	CPP0631	QUẢN LÝ BÁN HÀNG – 1	LỚP VÀ ĐỐI TƯỢNG	Lớp bạn - Kế thừa	3
</> 239	CPP0632	QUẢN LÝ BÁN HÀNG – 2	LỚP VÀ ĐỐI TƯỢNG	Lớp bạn - Kế thừa	3
</> 240	CPP0716	ĐẾM SỐ CÁCH DI CHUYỂN	CÁC BÀI TOÁN NÂNG CAO	Kỹ thuật sinh kê tiếp và Quay lui	3
</> 241	CPP0720	DÂY SỐ BITONIC	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	3
</> 242	CPP0732	TỔNG LỚN NHẤT CỦA DÂY CON TĂNG DẦN	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	3



STT	Mã	Tiêu đề	Nhóm	Chủ đề con	Độ khó
</> 243	CPP0761	ƯỚC SỐ CHUNG LỚN NHẤT CỦA SỐ NGUYÊN LỚN	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	3
</> 245	CPP0140	SỐ HOÀN HẢO	LẬP TRÌNH C++ CƠ BẢN	Ước số và Ước số chung lớn nhất	4
</> 246	CPP0214	SỐ LỚN NHẤT CỦA DÃY CON LIÊN TỤC	MẢNG VÀ CON TRỎ	Mảng một chiều	4
</> 247	CPP0233	MA TRẬN XOẮN ỐC - 2	MẢNG VÀ CON TRỎ	Mảng hai chiều	4
</> 248	CPP0234	MA TRẬN XOẮN ỐC - 3	MẢNG VÀ CON TRỎ	Mảng hai chiều	4
</> 256	CPP0361	MẬT KHẨU	XỬ LÝ XÂU KÝ TỰ	Kiểu dữ liệu string và áp dụng	4
</> 257	CPP0532	DIỆN TÍCH ĐA GIÁC	CẤU TRÚC VÀ MẢNG CẤU TRÚC	Ứng dụng cấu trúc	4
</> 263	CPP0736	DÃY CON LIÊN TỤC NHỎ NHẤT	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	4
</> 265	CPP0744	BẢNG MÀU R - G - B	CÁC BÀI TOÁN NÂNG CAO	Ứng dụng toán học	4

Hello World

Bài làm tốt nhất

Viết chương trình in ra màn hình dòng chữ:

Hello PTIT.

### **Input**

Không có dữ liệu vào

### **Output**

Hello PTIT.

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH TỔNG 1 ĐẾN N

### Bài làm tốt nhất

Cho số nguyên dương N.

Hãy tính  $S = 1 + 2 + \dots + N$

#### Dữ liệu vào:

- Dòng đầu ghi số bộ test, không quá 10
- Mỗi dòng ghi một số nguyên dương N, không quá  $10^9$

#### Kết quả:

Với mỗi test, ghi kết quả trên một dòng.

#### Ví dụ:

Input	Output
2	55
10	210
20	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## CHỮ HOA – CHỮ THƯỜNG

### Bài làm tốt nhất

Cho một chữ cái bất kỳ. Hãy kiểm tra xem đó là chữ hoa hay chữ thường. Nếu là chữ thường thì in ra chữ hoa, nếu là chữ hoa thì in ra chữ thường tương ứng.

### Input

Dòng đầu ghi số bộ test. Mỗi bộ test có duy nhất một chữ cái.

### Output

Với mỗi bộ test, ghi ra kết quả trên một dòng.

### Ví dụ

Input	Output
2	b
B	Q
q	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH TỔNG PHÂN THỨC

### Bài làm tốt nhất

Viết chương trình tính tổng  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$

### Input

Dữ liệu vào chỉ có 1 dòng ghi số N không quá 6 chữ số.

### Output

Kết quả ghi chính xác với **4 số phần thập phân**.

### Ví dụ

Input	Output
2	1.5000

**Chú ý:** Giới hạn thời gian chạy là 5 giây.

Giới hạn thời gian: 5s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH TỔNG GIAI THỪA

### Bài làm tốt nhất

Viết chương trình tính tổng  $S = 1 + 1.2 + 1.2.3 + \dots + 1.2.3 \dots N$

### Input

Dữ liệu vào chỉ có 1 dòng ghi số N không quá 20.

### Output

Kết quả ghi trên một dòng.

### Ví dụ

Input	Output
3	9

Giới hạn thời gian: 1 s

Giới hạn bộ nhớ: 65536 Kb

## **SỐ LỘC PHÁT**

### **Bài làm tốt nhất**

Một số được gọi là “lộc phát” nếu chỉ có các chữ số 0,6,8. Nhập vào một số nguyên dương không quá 9 chữ số, hãy kiểm tra xem đó có phải số lộc phát hay không.

### **Input**

Dòng đầu là số bộ test, mỗi dòng tiếp theo ghi một số cần kiểm tra.

### **Output**

Nếu đúng in ra YES, nếu sai in ra NO.

### **Ví dụ**

Input	Output
3	NO
181	YES
88888	YES
608868	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## **SỐ THUẬN NGHỊCH**

### **Bài làm tốt nhất**

Một số nguyên dương được gọi là số thuận nghịch nếu viết theo chiều ngược lại vẫn có cùng giá trị với số ban đầu. Ví dụ số 121 là số thuận nghịch, số 123 không phải số thuận nghịch.

Viết chương trình kiểm tra số thuận nghịch

### **Input**

Dòng đầu ghi số bộ test, mỗi bộ test ghi một số nguyên dương không quá 18 chữ số

### **Output**

Với mỗi bộ test, nếu đúng ghi ra YES, nếu sai ghi ra NO.

### **Ví dụ**

<b>Input</b>	<b>Output</b>
2	NO
123456789012345	YES
12345678987654321	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb



## CHẤM ĐIỂM TRẮC NGHIỆM

### Bài làm tốt nhất

Trong những năm gần đây, hình thức thi trắc nghiệm được sử dụng phổ biến trong kỳ thi Trung học phổ thông quốc gia. Ưu điểm của hình thức thi trắc nghiệm là có thể chấm tự động hoàn toàn mà không cần sự can thiệp của con người, điều này đảm bảo tính khách quan và chính xác. Giả sử có 2 mã đề thi 101 và 102 có các đáp án được cho trong bảng dưới đây:

Câu	Đề 101	Đề 102
1	A	A
2	B	C
3	B	C
4	A	A
5	D	B
6	C	C
7	C	D
8	A	D
9	B	B
10	D	B
11	C	C
12	C	D
13	A	D
14	B	B
15	D	B

Nhiệm vụ của bạn là chấm điểm cho bài thi của các thí sinh trong một phòng thi. Điểm thi được chấm trên thang điểm 10, làm tròn đến 2 chữ số phần thập phân.

### Input:

Dòng đầu tiên đưa vào số bài thi T.

Các dòng tiếp theo mỗi đưa vào mã đề thi và các đáp án làm bài của thí sinh cách nhau bởi một khoảng trống. Giả sử tất cả các đáp án đều được điền đầy đủ.

T thỏa mãn điều kiện  $0 < T \leq 1000$

### Output:

Điểm thi được làm tròn đến hai chữ số phần thập phân.

Input	Output
1	10.00
101 A B B A D C C A B D C C A B D	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CÂN BẰNG CHẴN LẺ

### Bài làm tốt nhất

Một số được gọi là “cân bằng chẵn lẻ” nếu số chữ số chẵn và số chữ số lẻ là bằng nhau. Tất nhiên khi đó số chữ số của nó phải là chẵn.

Cho số  $N$  là một số chẵn ( $1 < N < 7$ ). Hãy liệt kê các số cân bằng chẵn lẻ có  $N$  chữ số. Mỗi dòng ghi ra 10 số thỏa mãn.

### Input

Chỉ có duy nhất số  $N$  (chẵn)

### Output

Ghi ra các số cân bằng chẵn lẻ có  $N$  chữ số theo thứ tự từ nhỏ đến lớn. Mỗi dòng ghi 10 số.

### Ví dụ

Input	Output
2	10 12 14 16 18 21 23 25 27 29 30 32 34 36 38 41 43 45 47 49 50 52 54 56 58 61 63 65 67 69 70 72 74 76 78 81 83 85 87 89 90 92 94 96 98

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## MÃ SỐ QUỐC GIA

### Bài làm tốt nhất

Trong mã hàng hóa người ta thường ghi kèm theo mã số quốc gia sản xuất. Nếu sản xuất tại Việt Nam thì mã tương ứng là 084. Bài toán đặt ra là cho một dãy mã dạng số nguyên không quá 18 chữ số. Hãy loại bỏ đoạn mã 084 ra khỏi mã ban đầu.

Dữ liệu đảm bảo dãy mã luôn có duy nhất một lần cụm 084.

### Input

Dòng đầu ghi số bộ test. Mỗi test là một số nguyên có ít nhất 4 chữ số nhưng không quá 18 chữ số.

### Output

Ghi ra kết quả sau khi loại bỏ 084

### Ví dụ

Input	Output
3	123567
123084567	3300478845
3300478808445	1
1084	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ LIÊN KÈ

### Bài làm tốt nhất

Viết chương trình kiểm tra số nguyên dương N có thỏa mãn tính chất: tất cả các chữ số cạnh nhau chỉ sai khác nhau đúng một đơn vị hay không. Ví dụ: số 123212 là số thỏa mãn, số 34578 không thỏa mãn.

### Input

Dòng đầu ghi số số test (không quá 20). Mỗi test là 1 số nguyên dương N có ít nhất 2 chữ số, nhưng không quá 18 chữ số.

### Output

Ghi ra YES hoặc NO

### Ví dụ

Input	Output
3	YES
123212	YES
5654345676	NO
10233211123	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## KHOẢNG CÁCH

### Bài làm tốt nhất

Cho 2 điểm A và B trong không gian Oxy. Hãy tính khoảng cách giữa hai điểm đó.

#### Input

- Dòng đầu ghi số bộ test, không quá 20.
- Mỗi bộ test có 4 số thực lần lượt là tọa độ của 2 điểm A và B, giá trị tuyệt đối không quá 1000.

#### Output

Với mỗi bộ test, viết ra khoảng cách giữa 2 điểm với 4 chữ số sau dấu phẩy.

#### Ví dụ

Input	Output
2	5.0000
0 0 0 5	193.0648
0 199 5 6	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## SỐ MAY MẮN

### Bài làm tốt nhất

John rất thích con số 86 vì theo John đó là con số may mắn. Khi bắt gặp một số nguyên X John muốn kiểm tra xem trong dạng biểu diễn của nó kết thúc là số 86 hay không? Ví dụ số 111539786 kết thúc là số 86, còn số 123456789 thì không.

Nhiệm vụ của bạn là viết một chương trình đọc số nguyên X và kiểm tra xem trong dạng biểu diễn của nó kết thúc là số 86 hay không?

### Input

Dữ liệu vào gồm nhiều bộ dữ liệu tương ứng với nhiều test. Dòng đầu tiên chứa một số nguyên dương không lớn hơn 20 là số lượng các bộ dữ liệu. Các dòng tiếp theo chứa các bộ dữ liệu, các số không quá 9 chữ số.

### Output

Với mỗi bộ dữ liệu, ghi ra trên một dòng câu trả lời, ghi số 1 nếu trong dạng biểu diễn của nó kết thúc là số 86, ghi số 0 trong trường hợp ngược lại.

### Ví dụ

#### Input:

3

111539786

123456789

8686

#### Output:

1

0

1

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 2048 Kb

## PHÂN TÍCH THỪA SỐ NGUYÊN TỐ - 1

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy đưa ra tất cả các ước số nguyên tố của  $N$  cùng lũy thừa của nó. Ví dụ  $N=100 = 2^2 \times 5^2$ .  $N = 35 = 5^1 \times 7^1$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên  $N$ .
- $T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10000$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 2 5 2
100	5 1 7 1
35	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## TÍNH TỔNG CHỮ SỐ

### Bài làm tốt nhất

Thầy Tí rất thích thú với những con số, cô giáo đã giao cho Tí một bài tập về rút gọn các con số. Phép rút gọn được thực hiện như sau: từ một số ban đầu, số mới được tạo thành bằng cách cộng các chữ số của số ban đầu với nhau. Sau đó Tí phải thực hiện tiếp tục với con số vừa mới thu được.

Quá trình rút gọn kết thúc khi số thu được chỉ có duy nhất 1 chữ số.

Các bạn hãy cùng Tí đi tìm chữ số cuối cùng của phép rút gọn!

Input

Dòng đầu tiên gồm số lượng test  $T$  ( $T \leq 100$ ).

$T$  dòng tiếp theo, mỗi dòng gồm một số nguyên dương ( $\leq 10^9$ ).

Output

Hãy in ra chữ số cuối cùng sau khi thực hiện phép rút gọn.

Example

**Input:**

3  
1009  
167  
102

**Output:**

1  
5  
3

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ CHUNG - BỘI SỐ CHUNG

### Bài làm tốt nhất

Cho hai số nguyên  $a, b$ . Nhiệm vụ của bạn là tìm bội số chung nhỏ nhất và ước số chung lớn nhất của  $a$  và  $b$ . Bội số chung nhỏ nhất của  $a$  và  $b$  ký hiệu là  $\text{LCM}(a, b)$  và ước số chung lớn nhất của  $a$  và  $b$  ký hiệu là  $\text{GCD}(a, b)$ .

Input:

- Dòng đầu tiên đưa vào  $T$  là số lượng bộ test.
- $T$  dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một cặp số  $a, b$  được viết cách nhau một vài khoảng trống.
- $T, a, b$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq a, b \leq 10^8$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	10 5
5 10	56 2
14 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ CHUNG LỚN NHẤT CỦA N SỐ NGUYÊN DƯƠNG ĐẦU TIÊN

### Bài làm tốt nhất

Cho số tự nhiên  $n$ . Nhiệm vụ của bạn là tìm số nguyên nhỏ nhất chia hết cho  $1, 2, \dots, n$ .

Input:

- Dòng đầu tiên đưa vào  $T$  là số lượng bộ test.
- $T$  dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một số tự nhiên  $n$ .
- $T$  thỏa mãn ràng buộc:  $1 \leq T \leq 10^4$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	6
3	60
5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## KIỂM TRA NGUYÊN TỐ

### Bài làm tốt nhất

*Một số được gọi là số nguyên tố nếu nó chỉ có 2 ước là 1 và chính nó. Số 0 và 1 không được coi là số nguyên tố.*

**Yêu cầu:** Cho số  $n$ , hãy kiểm tra xem  $n$  có là số nguyên tố hay không.

### Dữ liệu:

Một dòng duy nhất chứa số  $n$  ( $0 \leq n \leq 10^9$ )

### Kết quả:

In ra “YES” nếu  $n$  là số nguyên tố, và “NO” trong trường hợp còn lại.

Input	Output
2	YES

Input	Output
4	NO

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## PHÂN TÍCH THỪA SỐ NGUYÊN TỐ - 2

### Bài làm tốt nhất

Cho số nguyên dương  $n$  ( $2 \leq n \leq 10^9$ ), hãy phân tích  $n$  ra thừa số nguyên tố.

#### Dữ liệu:

Một dòng duy nhất chứa số  $n$ .

#### Kết quả:

Mỗi dòng ghi một thừa số nguyên tố và số mũ tương ứng cách nhau bởi dấu cách.

Các thừa số nguyên tố in ra theo thứ tự tăng dần.

#### Ví dụ:

Input	Output
4	2 2

Input	Output
168	2 3 3 1 7 1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ SỐ NGUYÊN TỐ - 1

### Bài làm tốt nhất

Viết chương trình cho phép nhập vào hai số nguyên dương và tìm tất cả các số nguyên tố nằm trong khoảng đó.

#### Input

Chỉ có 2 số nguyên dương a và b (không quá  $10^6$ )

#### Output

Ghi ra lần lượt các số nguyên tố trong khoảng. Cách nhau một khoảng trống.

#### Ví dụ

Input	Output
10 50	11 13 17 19 23 29 31 37 41 43 47

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ SỐ NGUYÊN TỐ - 2

### Bài làm tốt nhất

Hãy sinh ra tất cả các số nguyên tố trong khoảng  $[M, N]$ . Ví dụ  $M=1, N=10$  ta có kết quả 2 3 5 7.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp mỗi dòng đưa vào một bộ test. Mỗi bộ test là bộ đôi  $M, N$  được viết cách nhau một vài khoảng trống.
- $T, M, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq M \leq N \leq 10000; N-M \leq 10000$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 3 5 7
1 10	3 5
3 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CẶP SỐ NGUYÊN TỐ ĐẦU TIÊN CÓ TỔNG BẰNG N

### Bài làm tốt nhất

Cho số tự nhiên N. Hãy tìm cặp số nguyên tố đầu tiên có tổng là N. Nếu không tồn tại cặp số nguyên tố có tổng bằng N, hãy đưa ra -1.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm là một số N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^6$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 2
4	
8	3 5

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## ƯỚC SỐ NGUYÊN TỐ

### Bài làm tốt nhất

Cho số nguyên dương  $N$ . Hãy đưa ra tất cả các ước số nguyên tố của  $N$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một số nguyên dương  $N$  được ghi trên một dòng.
- $T, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $2 \leq N \leq 10^{10}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3 3 5 7
315	31
31	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ NGUYÊN TỐ LỚN NHẤT

### Bài làm tốt nhất

Cho số nguyên dương  $N$ . Hãy đưa ra ước số nguyên tố lớn nhất của  $N$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một số nguyên dương  $N$  được ghi trên một dòng.
- $T, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $2 \leq N \leq 10^{10}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
315	31
31	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ NGUYÊN TỐ NHỎ HƠN N

### Bài làm tốt nhất

Cho số nguyên dương  $N$ . Hãy đưa ra tất cả các số nguyên tố nhỏ hơn hoặc bằng  $N$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một số nguyên dương  $N$  được ghi trên một dòng.
- $T, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $2 \leq N \leq 10^4$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 3 5 7
10	2 3 5 7 11 13 17 19 23 29 31
35	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ NGUYÊN TỐ THỨ K

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy đưa ra ước số nguyên tố thứ  $k$  của  $N$ . Đưa ra  $-1$  nếu không tồn tại ước số thứ  $k$  của  $N$ . Ví dụ  $N = 225$ ,  $k = 2$  ta có kết quả là  $3$  vì  $225 = 3 \times 3 \times 5 \times 5$ . Với  $N = 81$ ,  $k = 5$  ta có kết quả  $-1$  vì  $81 = 3 \times 3 \times 3 \times 3$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một bộ đôi  $N$  và  $k$ .
- $T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N$ ,  $k \leq 10^4$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
225 2	-1
81 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ SỐ CÓ BA ƯỚC SỐ

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy liệt kê tất cả các số có đúng ba ước số. Ví dụ  $n=100$ , ta có các số 4, 9, 25, 49.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số  $N$ .
- $T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^6$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 9 25 49
50	4 9 25 49 121 169
200	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CẶP SỐ NGUYÊN TỐ

### Bài làm tốt nhất

Cho số nguyên dương  $N > 2$ . Hãy đưa ra cặp số nguyên tố  $p, q$  đầu tiên tìm được có tổng đúng bằng  $N$ . Ví dụ  $N = 6$  ta có cặp số nguyên tố đầu tiên là  $3 + 3 = 6$ .

Trong trường hợp không tìm thấy đáp án thì không in ra gì cả.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số chẵn  $N$ .
- $T, N$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 4 \leq N \leq 10000$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng. Nếu không có cặp số nào thỏa mãn thì không in ra gì cả.

Input:	Output:
2	3 71
74	3 1021
1024	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ FIBONACCI THỨ N

### Bài làm tốt nhất

Dãy số Fibonacci được định nghĩa theo công thức như sau:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2} \text{ với } n > 2$$

Viết chương trình tính số Fibonacci thứ n (với n không quá 92)

**Dữ liệu vào:** Dòng đầu ghi số bộ test. Mỗi bộ test là một số nguyên n.

**Kết quả:** Với mỗi bộ test, ghi ra số Fibonacci thứ n trên một dòng.

**Ví dụ**

Input	Output
3	1
2	5
5	6765
20	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## CHIA DƯ

### Bài làm tốt nhất

Cho hai số nguyên dương  $a$  và  $m$ . Nhiệm vụ của bạn là tìm  $x$  nhỏ nhất trong khoảng  $[0, m-1]$  sao cho  $a * x \equiv 1 \pmod{m}$ . Ví dụ  $a = 3, m = 11$  ta tìm được  $x = 4$  vì  $4 * 3 \% 11 = 1$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi  $a, m$  được viết cách nhau một vài khoảng trống.
- $T, a, m$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 1 \leq a \leq m \leq 100$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng. Nếu phương trình đồng dư không có nghiệm, hãy đưa ra -1

Input:	Output:
2	4
3 11	12
10 17	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## CHIA DƯ TỪ 1 ĐẾN N

### Bài làm tốt nhất

Cho hai số nguyên không âm  $N$  và  $K$ . Nhiệm vụ của bạn là tìm  $S = 1\%K + 2\%K + \dots + N\%K$ . Ví dụ với  $N = 10, K=2$  ta có  $S = 1\%2 + 2\%2 + 3\%2 + 4\%2 + 5\%2 + 6\%2 + 7\%2 + 8\%2 + 9\%2 + 10\%2 = 5$ . Yêu cầu độ phức tạp thuật toán là hằng số

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi  $N, K$  được viết cách nhau một vài khoảng trống.
- $T, N, K$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 0 \leq N \leq 1000; 0 \leq K \leq 10^{12}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	55
10 55	1
1 11	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TỔNG CHIA DƯ CHO K

### Bài làm tốt nhất

Cho hai số nguyên không âm  $N$  và  $K$ . Nhiệm vụ của bạn là kiểm tra xem  $K = 1\%K + 2\%K + \dots + N\%K$  hay không. Đưa ra 1 hoặc 0 nếu cặp  $N, K$  thỏa mãn hoặc không thỏa mãn yêu cầu bài toán. Ví dụ với  $N = 10, K=55$  ta có kết quả là 1 vì  $55 = 1\%55 + 2\%55 + 3\%55 + \dots + 10\%55$ . Ngược lại,  $N=4, K=11$  có kết quả là 0 vì  $11 \neq 1\%11 + 2\%11 + 3\%11 + 4\%11$ .

#### Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi  $N, K$  được viết cách nhau một vài khoảng trống.
- $T, N, K$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 0 \leq N \leq 1000; 0 \leq K \leq 10^{12}$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
10 55	0
1 11	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CHÊNH LỆCH NHỎ NHẤT

### Bài làm tốt nhất

Cho dãy số  $A[]$  gồm có  $N$  phần tử. Bạn cần tìm chênh lệch nhỏ nhất giữa hai phần tử bất kì trong dãy số đã cho.

#### Input:

- Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).
- Mỗi test gồm số nguyên  $N$  ( $1 \leq N \leq 100\,000$ ).
- Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^9$ ).

#### Output:

- Với mỗi test, in ra trên một dòng là đáp án tìm được.

#### Ví dụ:

Input:	Output
3	1
6	4
1 5 3 19 18 25	1
4	
30 5 20 9	
7	
1 19 2 31 38 25 100	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## KHOẢNG CÁCH NHỎ NHẤT

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số chưa được sắp xếp. Hãy tìm  $\text{Min}(A[i]-A[j]) : i \neq j$  và  $i, j = 0, 1, 2, \dots, n-1$ . Ví dụ với  $A[] = \{1, 5, 3, 19, 18, 25\}$  ta có kết quả là  $1 = 19-18$ . với  $A[] = \{1, 19, -4, 31, 28, 35, 100\}$  ta có kết quả là  $3 = 31-28$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $-10^3 \leq A[i] \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5	6
2 4 5 7 9	
10	
87 32 99 75 56 43 21 10 68 49	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ NHỎ NHẤT CHƯA XUẤT HIỆN

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên bao gồm cả số 0. Nhiệm vụ của bạn là tìm số nguyên dương nhỏ nhất không có mặt trong mảng. Ví dụ với mảng  $A[] = \{5, 8, 3, 7, 9, 1\}$ , ta có kết quả là 2.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^6$ ;  $-10^6 \leq A[i] \leq 10^6$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6
5	2
1 2 3 4 5	
5	
0 -10 1 3 -20	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **DÃY TAM GIÁC**

### **Bài làm tốt nhất**

Cho mảng  $A[]$  gồm  $n$  số được thiết lập theo nguyên tắc nửa đầu tăng dần nửa sau giảm dần. Hãy tìm số lớn nhất của mảng. Ví dụ với mảng  $A[] = \{1, 2, 3, 4, 5, 2, 1\}$ , ta có kết quả 5.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^7$ ;  $0 \leq A[i] \leq 10^7$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 5 1 2 7 4 3	7

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## PHẦN TỬ LỚN NHẤT

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Hãy tìm phần tử lớn nhất của mảng. Ví dụ với mảng  $A[] = \{7, 10, 4, 3, 20, 15\}$  ta nhận được kết quả là 20.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^5$ ;  $1 \leq A[i] \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	20
6	12
7 10 4 3 20 15	
6	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## QUAY VÒNG DÂY SỐ 1

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử và số  $d$ . Hãy thực hiện phép quay vòng  $d$  phần tử của mảng  $A[]$ . Ví dụ với mảng  $A[] = \{1, 2, 3, 4, 5\}$ ,  $d = 2$  thì ta có kết quả  $A[] = \{3, 4, 5, 1, 2\}$ .

**Input:**

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là hai số  $n$  và  $d$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, d, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 10$ ;  $1 \leq d < n \leq 10^6$ ;  $1 \leq A[i] \leq 10^5$ ;

**Output:**

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3 4 5 1 2
5 2	5 6 7 1 2 3 4
1 2 3 4 5	
7 4	
1 2 3 4 5 6 7	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## PHẦN TỬ NHỎ NHẤT THỨ K

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số và số  $k$ . Hãy tìm phần tử nhỏ nhất thứ  $k$  của mảng. Ví dụ với mảng  $A[] = \{7, 10, 4, 3, 20, 15\}$ ,  $k=3$  ta nhận được số nhỏ nhất thứ  $k$  là 7.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$  và số  $k$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq k \leq n \leq 10^5$ ;  $1 \leq A[i] \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
6 3	8
7 10 4 3 20 15	
6 4	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH TỔNG TRONG KHOẢNG

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử và  $Q$  câu hỏi. Mỗi câu hỏi  $Q$  là bộ đôi hai số  $L$  và  $R$ . Nhiệm vụ của bạn là tìm tổng các phần tử của mảng  $A[]$  của mỗi câu hỏi  $Q$ . Ví dụ với mảng  $A[] = \{1, 1, 2, 1, 3, 4, 5, 2, 8\}$ , các câu hỏi  $Q$ :  $[1, 5]$ ,  $[2, 4]$ ,  $[3, 5]$  ta sẽ có các câu trả lời : 8 , 4, 6.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm ba phần: phần thứ nhất đưa vào  $n$ ,  $Q$  là số phần tử của mảng  $A[]$  và số lượng câu hỏi  $Q$ ; phần tiếp theo đưa vào  $n$  số  $A[i]$  của mảng; phần cuối cùng đưa vào  $Q$  câu hỏi, mỗi câu hỏi là một bộ đôi  $L, R$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, Q, L, R, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq L \leq R \leq n$ ,  $Q \leq 10^4$ ;  $1 \leq A[i] \leq 10^3$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	8
9 3	
1 1 2 1 3 4 5 2 8	4
1 5	6
2 4	
3 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## BIẾN ĐỔI NHỊ PHÂN

### Bài làm tốt nhất

Cho ma trận  $A[N][M]$  chỉ bao gồm các số 0 và 1. Hãy sửa đổi các phần tử của ma trận  $A[][]$  theo nguyên tắc: nếu phần tử  $A[i][j] = 1$  ta thay tất cả các phần tử của hàng  $i$ , cột  $j$  bởi 1. Ví dụ với ma trận dưới đây sẽ minh họa cho phép biến đổi:

1	0	0	1	1	1	1	1
0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	1

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số  $N, M$ ; dòng tiếp là  $N \times M$  các phần tử của ma trận  $A[][]$ ; các phần tử được viết cách nhau một vài khoảng trống.
- $T, N, M$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N, M \leq 100$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	0 0 1
2 3	
0 0 0	1 1 1
0 0 1	1 1 1 1
3 4	1 1 1 1
1 0 0 1	1 0 1 1
0 0 1 0	
0 0 0 0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## BIÊN CỦA MA TRẬN

### Bài làm tốt nhất

Cho ma trận vuông  $A[N][N]$ . Hãy in các phần tử thuộc vùng biên.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $N$  là cấp của ma trận  $A[N][N]$ ; dòng tiếp theo đưa vào  $N \times N$  số  $A[i][j]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, N, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 100$ ;  $1 \leq A[i][j] \leq 150$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 4
4	5       8
1 2 3 4 5 6 7 8 1 2 3 4 5 6 7	1       4
8	5 6 7 8
3	45 48 54
45 48 54 21 89 87 70 78 15	21    87
	70 78 15

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## IN MA TRẬN - 1

### Bài làm tốt nhất

Cho ma trận vuông  $A[N][N]$ . Hãy in các phần tử thuộc theo hình con rắn.

• Hãy in các phần tử thuộc:



#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $N$  là cấp của ma trận  $A[N][N]$ ; dòng tiếp theo đưa vào  $N \times N$  số  $A[i][j]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, N, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 100$ ;  $1 \leq A[i][j] \leq 150$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	45 48 54 87 89 21 70 78 15
3	25 27 21 23
45 48 54 21 89 87 70 78 15	
2	
25 27 23 21	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## MA TRẬN NHỊ PHÂN

### Bài làm tốt nhất

Cho ma trận  $A[]$  có  $N$  hàng và 3 cột, trong đó các vị trí là các giá trị nhị phân (0 hoặc 1).  
Hãy đếm xem có bao nhiêu hàng mà số lượng số 1 nhiều hơn số lượng số 0.

### Input

Dòng đầu ghi số nguyên dương  $N$  (không quá 1000).

$N$  dòng tiếp theo, mỗi dòng ghi 3 giá trị nhị phân.

### Output

Ghi ra số dòng mà số lượng số 1 nhiều hơn số lượng số 0.

### Ví dụ

Input	Output
3  1 1 0  1 1 1  1 0 0	2
2  1 0 0  0 1 1	1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ SỐ KHÁC NHAU

### Bài làm tốt nhất

Cho dãy số  $A[]$  chỉ bao gồm các số nguyên dương không quá 1000. Hãy liệt kê các số khác nhau trong dãy theo thứ tự tăng dần.

### Input

Dòng đầu ghi số  $n$  là số phần tử của dãy (không quá 100).

Dòng thứ 2 ghi  $n$  số của mảng.

### Output

Ghi ra các số khác nhau trong dãy  $A[]$  theo thứ tự tăng dần, mỗi số cách nhau một khoảng trống.

### Ví dụ

Input	Output
10	3 6 7 9 13
3 7 9 3 6 13 7 7 9 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍCH MA TRẬN

### Bài làm tốt nhất

Viết chương trình tính tích hai ma trận A cỡ  $n \times m$  và ma trận B cỡ  $m \times p$ .

Với  $1 < n, m, p < 50$ . Các giá trị trong ma trận đều nguyên dương và không vượt quá 1000.

### Input

Dòng đầu ghi 3 số  $n, m, p$

$n$  dòng tiếp theo ghi ma trận A

$m$  dòng tiếp theo ghi ma trận B

### Output

Ghi ra ma trận tích

### Ví dụ

Input	Output
3 4 3	30 30 30
1 2 3 4	21 21 21
4 2 3 1	25 25 25
2 4 1 3	
1 1 1	
2 2 2	
3 3 3	
4 4 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## ĐẾM SỐ PHẦN TỬ LẶP LẠI

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $N$  phần tử. Hãy đếm số phần tử bị lặp lại ít nhất 1 lần. Ví dụ với mảng  $A[] = \{5, 6, 1, 2, 1, 4\}$  thì ta có đáp án là 2 vì có 2 phần tử 1.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào số phần tử của mảng  $N$ ; dòng tiếp theo là  $N$  số  $A[i]$  là các phần tử của mảng  $A[]$ .
- $T, N, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^6$ ,  $1 \leq A[i] \leq 10^6$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
5	
4 5 1 2 1	4
6	
10 20 30 30 20 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## KÝ TỰ KHÔNG LẶP

### Bài làm tốt nhất

Cho xâu ký tự S. Nhiệm vụ của bạn là in ra tất cả các ký tự không lặp khác nhau trong S. Ví dụ S = "ABCDEABC" ta nhận được kết quả là "DE".

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq \text{Length}(S) \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng. In ra theo thứ tự chữ cái xuất hiện trong xâu ban đầu.

Input:	Output:
2 ABCDEAABC ABC	DE ABC

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẾM TỪ

### Bài làm tốt nhất

Một từ được hiểu là dãy các ký tự liên tiếp không chứa ký tự space, '\t', '\n', Cho xâu ký tự S có các ký tự space, '\t', '\n', hãy đếm số các từ của S.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq \text{length}(S) \leq 10^3$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5
Print the number of words	9
Print the number of words present in the string	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## XÂU PANGRAM

### Bài làm tốt nhất

Một chuỗi ký tự được gọi là Pangram nếu nó chứa đầy đủ các ký tự từ 'a',..'z'. Cho chuỗi ký tự S và số K. Hãy xác định xem có thể thực hiện nhiều nhất K phép biến đổi các ký tự để S trở thành Pangram hay không? Mỗi phép biến đổi là một phép thay thế ký tự này bằng một ký tự khác.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào chuỗi ký tự S; dòng tiếp theo đưa vào số K.
- T, S, K thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq K \leq 62$ ;  $1 \leq \text{Length}(S) \leq 10^6$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 qwqqwqeqqwsdadsdasadsfsdsdsdasasas	0
4 qwqqwqeqqwsdadsdasadsfsdsdsdasasas	1
24	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LOẠI BỎ TỪ TRONG XÂU

### Bài làm tốt nhất

Viết chương trình cho phép nhập vào một chuỗi và từ cần loại bỏ khỏi chuỗi. Thực hiện loại bỏ từ và in ra kết quả

Trong đó:

#### INPUT

- Hàng thứ nhất là chuỗi ban đầu
- Hàng tiếp theo là từ cần loại bỏ

#### OUTPUT

- Chuỗi kết quả

Input	Output
Toi Yeu PTIT	Yeu PTIT
Toi	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CHÚC MỪNG NĂM MỚI

### Bài làm tốt nhất

Tí năm nay đã lên lớp 1 rồi, Tết đến Tí rất vui vì nhận được rất nhiều lời chúc.

Vì mới tập viết nên Tí đã ghi lại tất cả các lời chúc đó. Cũng vì rất trân trọng các lời chúc nên Tí đã ghi tất cả các lời chúc bằng chữ IN HOA, tuy nhiên do mới tập viết nên Tí ghi không có dấu. Giờ ngồi lật lại cuốn nhật ký ghi các lời chúc, Tí thấy mình đã ghi được  $n$  lời chúc.

Tí muốn biết có bao nhiêu lời chúc khác nhau (hai lời chúc được gọi là khác nhau nếu chúng có độ dài khác nhau hoặc tồn tại ít nhất một vị trí mà ký tự ở vị trí đó của hai lời chúc là khác nhau, hay nói cách khác, đó là hai xâu ký tự khác nhau). Bạn hãy lập chương trình giúp Tí đếm xem có bao nhiêu lời chúc khác nhau nhé.

### Input:

Dòng đầu chứa số nguyên dương  $n$  là số lời chúc Tí ghi được;

$n$  dòng tiếp theo, mỗi dòng chứa một xâu ký tự  $S$  là một lời chúc.

$n$ ,  $S$  thỏa mãn ràng buộc:  $1 \leq n \leq 10^4$ ; Các lời chúc  $S$  có độ dài không quá 30 ký tự gồm các chữ cái la tinh IN HOA 'A'...'Z' và dấu cách.

### Output:

Một số nguyên dương duy nhất là số lời chúc khác nhau.

### Ví dụ:

Input:	Output:
4  CHUC MUNG NAM MOI  HAPPY NEW YEAR  CHUC MUNG TUOI MOI  CHUC MUNG NAM MOI	3

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ ĐẸP

### Bài làm tốt nhất

Một số được coi là đẹp nếu đó là số thuận nghịch và chỉ toàn các chữ số chẵn. Viết chương trình đọc vào các số nguyên dương có không quá 500 chữ số và kiểm tra xem số đó có đẹp hay không.

#### Input:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương  $n$  không quá 500 chữ số.

#### Output:

Mỗi bộ test viết ra trên một dòng chữ YES nếu đó là số đẹp, chữ NO nếu ngược lại

#### Ví dụ

Input	Output
4	NO
123456787654321	YES
86442824468	YES
8006000444422220000222244440006008	NO
235365789787654324567856578654356786556	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## NHỎ NHẤT - LỚN NHẤT

### Bài làm tốt nhất

Cho số tự nhiên  $m$  và số nguyên  $s$  không âm. Nhiệm vụ của bạn là tìm số bé nhất và lớn nhất có  $m$  chữ số và tổng chữ số bằng  $s$ .

#### Input

Dòng đầu gồm 2 số  $m$  và  $s$  ( $1 \leq m \leq 100, 0 \leq s \leq 900$ ).

#### Output

In ra kết quả của bài toán.

Số đầu tiên là số bé nhất, số thứ hai là số lớn nhất. Nếu không có đáp án in ra "-1 -1".

#### Example

##### Input:

2 15

##### Output:

69 96

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb



## CHIA HẾT CHO 11

### Bài làm tốt nhất

Cho số tự nhiên  $N$ , hãy kiểm tra xem  $N$  có phải là số chia hết cho 11 hay không? Đưa ra 1 nếu  $N$  chia hết cho 11, trái lại đưa ra 0.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một số tự nhiên  $N$ .
- $T, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq N \leq 10^{1000}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
76945	0
363588395960667043875487	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CHIA HẾT CHO 5

### Bài làm tốt nhất

Cho số tự nhiên  $N$  được biểu diễn dưới dạng nhị phân. Hãy kiểm tra xem  $N$  có phải là số chia hết cho 5 hay không? Đưa ra “Yes” nếu  $N$  chia hết cho 5, trái lại đưa ra “No”.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một biểu diễn nhị phân của  $N$ .
- $T, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq N \leq 10^{1000}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	Yes
1010	Yes
10000101001	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐỊA CHỈ EMAIL - 1

### Bài làm tốt nhất

Địa chỉ email của cán bộ PTIT được cấp theo nguyên tắc ghép tên với chữ cái đầu tiên của họ và tên đệm. Viết chương trình cho phép tạo các địa chỉ email theo tên cán bộ

### Input

Chỉ có một xâu ký tự họ tên độ dài không quá 100.

### Output

Ghi ra địa chỉ email theo mẫu trong ví dụ.

### Ví dụ

Input	Output
Nguyen vAn nAM	namnv@ptit.edu.vn

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CHUẨN HÓA HỌ TÊN

### Bài làm tốt nhất

Tên người dùng sẽ được chuẩn hóa theo nguyên tắc tên được viết sau cùng, phân tách với phần tên đệm và tên bởi dấu phẩy. Các chữ cái nằm trong tên đều được viết hoa; chữ cái đầu tiên của tên đệm và họ được viết hoa, các chữ cái còn lại viết thường.

### Input

Chỉ có một xâu họ tên độ dài không quá 100

### Output

Ghi ra xâu họ tên đã chuẩn hóa theo mẫu trong ví dụ

### Ví dụ

Input	Output
ngUyeN vAN Nam	Nguyen Van, NAM

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẦU CUỐI GIỐNG NHAU

### Bài làm tốt nhất

Cho xâu ký tự S. Hãy đếm tất cả các xâu con của S có ký tự đầu và ký tự cuối giống nhau. Ví dụ với xâu “aba” ta có 4 xâu con bao gồm: “a”, “b”, “a”, “aba”.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq k \leq 26$ ;  $1 \leq \text{Length}(S) \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
abca	4
aba	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **DÃY ƯU THỂ**

### **Bài làm tốt nhất**

Cho dãy  $A[]$  chỉ bao gồm các số nguyên dương không quá  $10^5$  nhưng không biết trước số phần tử của dãy. Người ta gọi dãy  $A[]$  là dãy ưu thể nếu thỏa mãn 1 trong 2 điều kiện sau đây:

- Dãy gọi là ưu thể chẵn nếu số phần tử của dãy là chẵn và số lượng số chẵn trong dãy nhiều hơn số lượng số lẻ.
- Dãy gọi là ưu thể lẻ nếu số phần tử của dãy là lẻ và số lượng số lẻ trong dãy nhiều hơn số lượng số chẵn.

Hãy kiểm tra xem dãy  $A[]$  có phải là dãy ưu thể hay không.

### **Dữ liệu vào**

- Dòng đầu ghi số bộ test, không quá 10
- Mỗi bộ test là một dãy các số nguyên dương (không quá  $10^4$ ) và có không quá 200 số, các số cách nhau 1 khoảng trống, không biết trước số lượng phần tử.

### **Kết quả**

- Nếu dãy  $A[]$  thỏa mãn là dãy ưu thể thì in ra YES, nếu không in ra NO

### **Ví dụ**

Input	Output
2	YES
11 22 33 44 55 66 77	NO
23 34 45 56 67 78 89 90 121 131 141 151 161 171	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## MÃ HÓA

### Bài làm tốt nhất

Cho một chuỗi ký tự độ dài không quá 100 chỉ bao gồm các chữ cái in hoa. Người ta thực hiện mã hóa bằng cách đếm các ký tự cạnh nhau giống nhau và viết số lượng phía sau các chữ cái đó.

Ví dụ chuỗi AAEECCCGGGD thì được mã hóa thành A2E1C4G3D1

Với giả thiết không có ký tự nào xuất hiện nhiều hơn 9 lần liên tiếp. Hãy viết chương trình mã hóa chuỗi ký tự theo cách như trên.

### Input

Dòng đầu ghi số bộ test. Mỗi bộ test ghi chuỗi chữ cái in hoa không quá 100 ký tự. Không có ký tự nào xuất hiện nhiều hơn 9 lần liên tiếp.

### Output

Với mỗi test ghi ra kết quả mã hóa.

### Ví dụ

Input	Output
2	A8
AAAAAAAAA	A2E1C4G3D1
AAEECCCGGGD	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **LOẠI BỎ NGUYÊN ÂM**

### **Bài làm tốt nhất**

Cho một chuỗi ký tự S chỉ bao gồm các ký tự chữ cái và không có khoảng trống. Hãy loại bỏ các nguyên âm trong S.

Kết quả được viết ra dưới dạng chữ cái viết thường của các phụ âm có mặt trong S, trước mỗi phụ âm ghi một ký tự dấu chấm '.'.

Các nguyên âm bao gồm: 'A', 'E', 'I', 'O', 'U', 'Y' (cả viết hoa và viết thường).

### **Input**

Chỉ có chuỗi S, độ dài không quá 100.

### **Output**

Ghi ra chuỗi kết quả

### **Ví dụ**

<b>Input</b>	<b>Output</b>
HocVienCNBCVT	.h.c.v.n.c.n.b.c.v.t

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## SẮP XẾP 0 -1 - 2

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Các phần tử của mảng  $A[]$  chỉ bao gồm các số 0, 1, 2. Hãy sắp xếp mảng  $A[]$  theo thứ tự tăng dần. Ví dụ với  $A[] = \{0, 2, 1, 2, 0\}$  ta kết quả  $A[] = \{0, 0, 1, 2, 2\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$  các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq A[i] \leq 2$ ;  $1 \leq n \leq 10^6$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	0 0 1 2 2
5	0 0 1
0 2 1 2 0	
3	
0 1 0	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP XEN KẼ - 1

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên khác nhau. Hãy đưa ra các phần tử của mảng theo khuôn dạng lớn nhất, nhỏ nhất, lớn thứ hai, nhỏ thứ 2, ... Ví dụ với  $A[] = \{9, 7, 12, 8, 6, 5\}$  ta đưa ra : 12, 5, 9, 6, 8, 7.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7 1 6 2 5 3 4
7	9 1 8 2 7 3 6 4
7 1 2 3 4 5 6	
8	
1 6 9 4 3 7 8 2	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP CHỮ SỐ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Nhiệm vụ của bạn là đưa ra mảng đã được sắp xếp bao gồm các chữ số của mỗi phần tử trong  $A[]$ . Ví dụ  $A[] = \{110, 111, 112, 113, 114\}$  ta có kết quả là  $\{0, 1, 2, 3, 4\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng tiếp theo là  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^7$ ;  $0 \leq A[i] \leq 10^{16}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 3 4 8
3	1 2 3 4 6
131 11 48	
4	
111 222 333 446	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH TÍCH

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử và mảng  $B[]$  gồm  $m$  phần tử. Nhiệm vụ của bạn là tìm tích giữa phần tử lớn nhất của mảng  $A[]$  và phần tử nhỏ nhất của mảng  $B[]$ . Ví dụ  $A[] = \{5, 7, 112, 9, 3, 6, 2\}$ ,  $B[] = \{1, 2, 6, -1, 0, 9\}$  ta có kết quả là  $-9 = 9 * (-1)$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào  $n, m$  tương ứng với số phần tử của mảng  $A[]$  và  $B[]$ ; dòng tiếp theo là  $n$  số  $A[i]$ ; dòng cuối cùng là  $m$  số  $B[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A[i], B[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, m \leq 10^6$ ;  $-10^8 \leq A[i] \leq 10^8$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	-9
6 6	20
5 7 9 3 6 2	
1 2 6 -1 0 9	
6 6	
1 4 2 3 10 2	
4 2 6 5 2 9	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## ĐẾM CẶP PHẦN TỬ CÓ TỔNG BẰNG K

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử và số  $k$ . Đếm tất cả các cặp phần tử của mảng có tổng bằng  $k$ . Ví dụ  $A[] = \{1, 5, 3, 4, 2\}$ ,  $k = 7$  ta có kết quả là 2 cặp  $(3, 4)$ ,  $(5, 2)$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$  và  $k$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$  các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 100$ ;  $0 \leq k \leq 100$ ,  $0 \leq A[i] \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	0
5 0	3
1 5 4 1 2	
3 2	
1 1 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP ĐẶT DÃY SỐ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho  $A[i] = i$ . Nếu phần tử  $A[j]$  của có giá trị khác  $j$ , hãy đưa ghi vào  $-1$ . Ví dụ với mảng  $A[] = \{-1, -1, 6, 1, 9, 3, 2, -1, 4, -1\}$  ta có kết quả  $A[] = \{-1, 1, 2, 3, 4, -1, 1, -1, -1, 9\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^7$ ;  $1 \leq A[i] \leq 10^{18}$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	-1 1 2 3 4 -1 6 -1 -1 9
10	0 1 -1 3 -1 -1
-1 -1 6 1 9 3 2 -1 4 -1	
6	
0 -3 1 -2 3 -4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ 0 Ở CUỐI DẪY

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho các số 0 để ở cuối cùng, các phần tử khác không được bảo toàn thứ tự trước sau. Ví dụ với mảng  $A[] = \{1, 2, 0, 0, 0, 3, 6\}$  ta có kết quả  $A[] = \{1, 2, 3, 6, 0, 0, 0\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^7$ ;  $0 \leq A[i] \leq 10^{18}$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 6 0 0 0
7	1 2 3 0 0 0 0
1 2 0 0 0 3 6	
6	
0 1 0 2 0 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## GHÉP DÂY SỐ

### Bài làm tốt nhất

Cho  $k$  mảng mỗi mảng gồm  $n$  phần tử đã được sắp xếp. Hãy đưa ra kết quả là một dãy đã được sắp xếp. Ví dụ với  $k = 3$ ,  $n = 4$  và mảng

$$A[] = \{ \{1, 3, 5, 7\}, \\ \{2, 4, 6, 8\} \\ \{0, 9, 10, 11\} \\ \};$$

sẽ cho ta kết quả  $A[] = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất dòng thứ nhất đưa là hai số  $k, n$ ;  $k$  dòng tiếp, mỗi dòng gồm  $n$  số của mảng  $A[k][n]$ . Các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $1 \leq k \leq 10$ ;  $1 \leq A[i][j] \leq 10^6$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	0 1 2 3 4 5 6 7 8 9 10 11
3 4	
1 3 5 7	
2 4 6 8	
0 9 10 11	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## TRỘN HAI DÃY VÀ SẮP XẾP

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử và mảng  $B[]$  gồm  $m$  phần tử. Nhiệm vụ của bạn là hợp nhất hai mảng  $A[]$  và  $B[]$  để được một mảng mới đã được sắp xếp. Ví dụ  $A[] = \{10, 5, 15\}$ ,  $B[] = \{20, 3, 2\}$  ta có kết quả là  $C[] = \{2, 3, 5, 10, 15, 20\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào  $n, m$  tương ứng với số phần tử của mảng  $A[]$  và  $B[]$ ; dòng tiếp theo là  $n$  số  $A[i]$ ; dòng cuối cùng là  $m$  số  $B[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A[i], B[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, m \leq 10^6$ ;  $-10^8 \leq A[i] \leq 10^8$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	2 3 5 10 15 20
3 3	
10 5 15	
20 3 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## BỔ SUNG PHẦN TỬ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Gọi  $L, R$  là max và min các phần tử của  $A[]$ . Nhiệm vụ của bạn là tìm số phần tử cần thiết cần thêm vào mảng để mảng có đầy đủ các số trong khoảng  $[L, R]$ . Ví dụ  $A[] = \{5, 7, 9, 3, 6, 2\}$  ta nhận được kết quả là 2 tương ứng với các số còn thiếu là 4, 8.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$ , tương ứng với số phần tử của mảng  $A[]$ ; dòng tiếp theo là  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, A[i] \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5	0
4 5 3 8 6	
3	
2 1 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÌM KIẾM TUẦN TỰ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Hãy tìm vị trí của phần tử đầu tiên có giá trị  $X$  trong mảng  $A[]$ . Nếu không tìm thấy  $X$  hãy đưa ra  $-1$ .

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $n$  là số các phần tử của mảng  $A[]$  và số  $X$  cần tìm; dòng tiếp theo đưa vào  $n$  số  $A[i]$  ( $1 \leq i \leq n$ ) các số được viết cách nhau một vài khoảng trống.
- $T, n, A, X$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N, X, A[i] \leq 10^6$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
5 16	-1
9 7 2 16 4	
7 98	
1 22 57 47 34 18 66	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÌM KIẾM NHỊ PHẦN

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử đã được sắp xếp. Hãy đưa ra 1 nếu  $X$  có mặt trong mảng  $A[]$ , ngược lại đưa ra -1.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $n$ ,  $X$  là số các phần tử của mảng  $A[]$  và số  $X$  cần tìm; dòng tiếp theo đưa vào  $n$  số  $A[i]$  ( $1 \leq i \leq n$ ) các số được viết cách nhau một vài khoảng trống.
- $T$ ,  $n$ ,  $A$ ,  $X$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N$ ,  $X$ ,  $A[i] \leq 10^6$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5 16	-1
9 7 2 16 4	
7 98	
1 22 57 47 34 18 66	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## SỐ NHỎ NHẤT CÒN THIẾU

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n-1$  phần tử bao gồm các khác nhau từ  $1, 2, \dots, n$ . Hãy tìm số nguyên dương nhỏ nhất không có mặt trong mảng  $A[]$ .

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $n$ ; dòng tiếp theo đưa vào  $n-1$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N$ ,  $A[i] \leq 10^7$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
5	9
1 2 3 5	
10	
1 2 3 4 5 6 7 8 10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÌM KIẾM TRONG DÂY SẮP XẾP VÒNG

### Bài làm tốt nhất

Một mảng được sắp được chia thành hai đoạn tăng dần được gọi là mảng sắp xếp vòng. Ví dụ mảng  $A[] = \{ 5, 6, 7, 8, 9, 10, 1, 2, 3, 4 \}$  là mảng sắp xếp vòng. Cho mảng  $A[]$  gồm  $n$  phần tử, hãy tìm vị trí của phần tử  $x$  trong mảng  $A[]$ .

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $n$  và  $x$ ; dòng tiếp theo đưa vào  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i], x$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N, x, A[i] \leq 10^7$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	9
10 3	3
5 6 7 8 9 10 1 2 3 4	
10 3	
1 2 3 4 5 6 7 8 9 10	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## SỐ NHỎ NHẤT VÀ NHỎ THỨ HAI

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử, hãy đưa ra số nhỏ nhất và số nhỏ thứ hai của mảng. Nếu không có số nhỏ thứ hai, hãy đưa ra  $-1$ .

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng tiếp theo đưa vào  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N$ ,  $A[i] \leq 10^7$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2
10	-1
5 6 7 8 9 10 1 2 3 4	
5	
1 1 1 1 1	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## TỔNG GẦN 0 NHẤT

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử, hãy tìm cặp phần tử có tổng gần nhất so với 0.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng tiếp theo đưa vào  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $2 \leq N \leq 10^3$ ,  $-10^6 \leq A[i] \leq 10^6$ .

#### Output:

- Đưa ra tổng gần nhất với 0 của cặp phần tử.

Input:	Output:
2	-68
3	-14
-8 -66 -60	
6	
-21 -67 -37 -18 4 -65	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb



## LIỆT KÊ K PHẦN TỬ LỚN NHẤT

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử, hãy tìm  $k$  phần tử lớn nhất của mảng. Các phần tử được đưa ra theo thứ tự giảm dần.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $N$  và  $K$ ; dòng tiếp theo đưa vào  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, N, K, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq K < N \leq 10^3$ ,  $1 \leq A[i] \leq 10^6$ .

#### Output:

- Đưa ra kết quả mỗi bộ test trên một dòng.

Input:	Output:
2	12 10 9
5 3	12 9
10 7 9 12 6	
6 2	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẾM SỐ LẦN XUẤT HIỆN

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử đã được sắp xếp. Hãy tìm số lần xuất hiện số  $X$  trong mảng. Nếu số lần xuất hiện số  $x$  trong mảng là 0 hãy đưa ra -1.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $N$  và  $X$ ; dòng tiếp theo đưa vào  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, N, X, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ,  $1 \leq A[i], X \leq 10^6$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
7 2	-1
1 1 2 2 2 3	
7 4	
1 1 2 2 2 3	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## PHẦN TỬ ĐẦU TIÊN LẶP LẠI

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $N$  phần tử. Hãy tìm phần tử lặp lại đầu tiên của mảng. Ví dụ với mảng  $A[] = \{5, 6, 1, 2, 1, 4\}$  thì ta có 1 là phần tử đầu tiên lặp lại trong mảng. Nếu không tồn tại đáp án, in ra -1.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào số phần tử của mảng  $N$ ; dòng tiếp theo là  $N$  số  $A[i]$  là các phần tử của mảng  $A[]$ .
- $T, N, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^6$ ,  $1 \leq A[i] \leq 10^6$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5	
4 5 1 2 1	30
6	
10 20 30 30 20 5 7	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## CẤU TRÚC ĐIỂM

### Bài làm tốt nhất

Viết chương trình khai báo cấu trúc điểm trong không gian hai chiều với hai tọa độ x và y (kiểu số thực double).

```
struct Point{  
  
    double x, y;  
  
}
```

Nhập hai điểm A và B. Tính khoảng cách giữa hai điểm A, B với độ chính xác 4 chữ số sau dấu phẩy.

### Input

- Dòng đầu ghi số bộ test, không quá 20.
- Mỗi bộ test có 4 số thực lần lượt là tọa độ của 2 điểm A và B, giá trị tuyệt đối không quá 1000.

### Output

Với mỗi bộ test, viết ra khoảng cách giữa 2 điểm với 4 chữ số sau dấu phẩy.

### Ví dụ

Input	Output
2	5.0000
0 0 0 5	193.0648
0 199 5 6	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++  
int main(){  
    struct Point A, B;  
    int t;
```

```
cin>>t;
while(t--){
    input(A); input(B);
    cout << fixed << setprecision(4) << distance(A,B) << endl;
}
return 0;
}
```

## CẤU TRÚC THÍ SINH

### Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Thí Sinh gồm các thông tin: Họ tên, Điểm môn 1, Điểm môn 2, Điểm môn 3 và Tổng điểm.

Đọc thông tin 1 thí sinh từ bàn phím và in ra màn hình 3 thông tin: Họ tên, Ngày sinh, Tổng điểm.

### Input

Gồm 5 dòng lần lượt, mỗi dòng ghi 1 thông tin: Họ tên, Ngày sinh, Điểm môn 1, Điểm môn 2, Điểm môn 3. Họ tên không quá 50 chữ cái, Ngày sinh viết đúng chuẩn dd/mm/yyyy. Các giá trị điểm là số thực (float).

### Output

Ghi ra Họ tên, Ngày sinh và Tổng điểm. Mỗi thông tin cách nhau một khoảng trống. Điểm được ghi ra với 1 số sau dấu phẩy.

### Ví dụ

Input	Output
Nguyen Hoang Ha 11/10/2001 4.5 10.0 5.5	Nguyen Hoang Ha 11/10/2001 20.0

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct ThiSinh A;
```

```
    nhap(A);  
    in(A);  
    return 0;  
}
```

## CẤU TRÚC PHÂN SỐ

### Bài làm tốt nhất

Viết chương trình xây dựng cấu trúc Phân số gồm hai phần tử là tử số và mẫu số. Các giá trị đều nguyên dương và không quá 18 chữ số.

Sau đó thực hiện nhập vào một phân số và in ra phân số đó ở dạng tối giản.

### Input

Có hai số nguyên dương lần lượt là tử số và mẫu số.

### Output

Ghi ra phân số tối giản như trong ví dụ

### Ví dụ

Input	Output
123 456	41 / 152

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main() {
    struct PhanSo p;
    nhap(p);
    rutgon(p);
    in(p);
    return 0;
}
```



## CẤU TRÚC SINH VIÊN

### Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float).

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

### Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
Nguyen Hoa Binh D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Hoa Binh D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++  
int main(){
```

```
    struct SinhVien a;  
    nhap(a);  
    in(a);  
    return 0;  
}
```

## CẤU TRÚC NHÂN VIÊN

### Bài làm tốt nhất

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán giá trị là 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nu
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập một nhân viên (không nhập mã) và in ra màn hình thông tin của nhân viên đó.

### Input

Gồm 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

### Output

Ghi ra đầy đủ thông tin nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trống.

### Ví dụ

Input
Nguyen Van Hoa
Nam
22/11/1982
Mo Lao-Ha Dong-Ha Noi
8333123456
31/12/2013
Output
00001 Nguyen Van Hoa Nam 22/11/1982 Mo Lao-Ha Dong-Ha Noi 8333123456 31/12/2013

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++  
int main(){  
    struct NhanVien a;  
    nhap(a);  
    in(a);  
    return 0;  
}
```

## TÍNH TỔNG HAI PHÂN SỐ

### Bài làm tốt nhất

Viết chương trình xây dựng cấu trúc Phân số gồm hai phần tử là tử số và mẫu số. Các giá trị đều nguyên dương và không quá 9 chữ số.

Sau đó thực hiện nhập vào hai phân số  $p$  và  $q$ . Tính tổng  $p + q$ , rút gọn và in ra kết quả.

### Input

Có bốn số nguyên dương lần lượt là tử số và mẫu số của  $p$  rồi đến  $q$ .

### Output

Ghi ra phân số tổng  $p + q$  ở dạng tối giản như trong ví dụ

### Ví dụ

Input	Output
123 456 12 34	1609/2584

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main() {
    struct PhanSo p,q;
    nhap(p); nhap(q);
    PhanSo t = tong(p,q);
    in(t);
    return 0;
}
```

## **DANH SÁCH NHÂN VIÊN**

### **Bài làm tốt nhất**

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nu
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã) và in ra màn hình danh sách nhân viên vừa nhập.

### **Input**

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

### **Output**

Ghi ra danh sách đầy đủ nhân viên theo đúng thứ tự nhập, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trống.

### **Ví dụ**

Input
3
Nguyen Van A
Nam
10/22/1982
Mo Lao-Ha Dong-Ha Noi
8333012345
31/12/2013
Ly Thi B

Nu

10/15/1988

Mo Lao-Ha Dong-Ha Noi

8333012346

22/08/2011

Hoang Thi C

Nu

04/02/1981

Mo Lao-Ha Dong-Ha Noi

8333012347

22/08/2011

## Output

00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013

00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011

00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    struct NhanVien ds[50];
    int N,i;
    cin >> N;
    for(i = 0; i < N; i++) nhap(ds[i]);
    inds(ds,N);
    return 0;
}
```

## KHAI BÁO LỚP SINH VIÊN - 1

### Bài làm tốt nhất

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

### Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
Nguyen Van A D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Van A D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++



```
int main(){
    SinhVien a;
    a.nhap();
    a.xuat();
    return 0;
}
```

## KHAI BÁO LỚP SINH VIÊN - 2

### Bài làm tốt nhất

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chồng toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

### Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
Nguyen Hoa Binh D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Hoa Binh D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    SinhVien a;
    cin >> a;
    cout << a;
    return 0;
}
```

## KHAI BÁO LỚP SINH VIÊN - 3

### Bài làm tốt nhất

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chồng toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy. Tên được đưa về dạng chuẩn.

### Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
NguyEn hoa Binh D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Hoa Binh D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++  
int main(){  
    SinhVien a;  
    cin >> a;  
    cout << a;  
    return 0;  
}
```

## KHAI BÁO LỚP PHÂN SỐ

### Bài làm tốt nhất

Viết chương trình xây dựng class Phân số.

Sau đó thực hiện nhập vào một phân số và in ra phân số đó ở dạng tối giản.

#### Input

Có hai số nguyên dương lần lượt là tử số và mẫu số. Các giá trị không quá 18 chữ số.

#### Output

Ghi ra phân số tối giản như trong ví dụ

#### Ví dụ

Input	Output
123 456	41/152

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main() {  
    PhanSo p(1,1);  
    cin >> p;  
    p.rutgon();  
    cout << p;  
    return 0;  
}
```

## KHAI BÁO LỚP NHÂN VIÊN

### Bài làm tốt nhất

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán giá trị là 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nu
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập một nhân viên (không nhập mã) trong đó có sử dụng chồng toán tử nhập/xuất và in ra màn hình thông tin của nhân viên đó.

### Input

Gồm 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

### Output

Ghi ra đầy đủ thông tin nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trống.

### Ví dụ

Input
Nguyen Van Hoa
Nam
11/22/1982
Mo Lao-Ha Dong-Ha Noi
8333123456
31/12/2013
Output
00001 Nguyen Van Hoa Nam 11/22/1982 Mo Lao-Ha Dong-Ha Noi 8333123456 31/12/2013

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++  
int main(){  
    NhanVien a;  
    a.nhap();  
    a.xuat();  
    return 0;  
}
```



## TÍNH TỔNG HAI ĐỐI TƯỢNG PHÂN SỐ

### Bài làm tốt nhất

Viết chương trình xây dựng class Phân số.

Sau đó thực hiện nhập vào hai phân số  $p$  và  $q$ . Tính tổng  $p + q$ , rút gọn và in ra kết quả.

#### Input

Có bốn số nguyên dương lần lượt là tử số và mẫu số của  $p$  rồi đến  $q$ . Các giá trị không quá 9 chữ số.

#### Output

Ghi ra phân số tổng  $p + q$  ở dạng tối giản như trong ví dụ

#### Ví dụ

Input	Output
1 2 3 4 5 6 1 2 3 4	1 6 0 9 / 2 5 8 4

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main() {
    PhanSo p(1,1), q(1,1);
    cin >> p >> q;
    cout << p + q;
    return 0;
}
```

## LIỆT KÊ XÂU NHỊ PHÂN

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Hãy đưa ra các xâu nhị phân có độ dài  $N$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số  $N$  được viết trên 1 dòng.
- $T, N$  thỏa mãn ràng buộc :  $1 \leq T, N \leq 20$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	00 01 10 11
2	000 001 010 011 100 101 110 111
3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## HOÁN VỊ LIÊN KỀ PHÍA TRƯỚC

### Bài làm tốt nhất

Cho số tự nhiên  $N$  và một hoán vị  $X[]$  của  $1, 2, \dots, N$ . Nhiệm vụ của bạn là đưa ra hoán vị trước đó của  $X[]$ . Ví dụ  $N=5$ ,  $X[] = \{1, 2, 3, 4, 5\}$  thì hoán vị trước đó của  $X[]$  là  $\{5, 4, 3, 2, 1\}$ .

### Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là số  $N$ ; dòng tiếp theo đưa vào hoán vị  $X[]$  của  $1, 2, \dots, N$ .
- $T, N, X[]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ .
- Input đảm bảo không có trường hợp hoán vị đã cho là đầu tiên (tức là luôn có hoán vị trước nó)

### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

### Ví dụ:

Input	Output
2	1 2 3 4 5
5	5 4 3 1 2
1 2 3 5 4	
5	
5 4 3 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍCH GIAI THỪA CÁC CHỮ SỐ

### Bài làm tốt nhất

Axe chơi một trò chơi với Lina.

Họ định nghĩa hàm  $F(x)$  với số  $x$  nguyên dương là tích giai thừa các chữ số của  $x$ .

Ví dụ  $F(135) = 1! * 3! * 5! = 720$ .

Đầu tiên, họ chọn một số  $a$  có  $n$  chữ số và có ít nhất một chữ số lớn hơn 1, có thể có chữ số không ở đầu. Sau đó họ tìm một số nguyên dương  $x$  lớn nhất thỏa mãn:

1.  $x$  không chứa chữ số 0 hoặc 1
2.  $F(x) = F(a)$

Hãy giúp Axe và Lina tìm ra được số đó.

### Input

Dòng đầu tiên chứa số bộ test  $T$  ( $T < 100$ ).

Mỗi test gồm một dòng chứa số  $n$  và số  $a$  ( $1 \leq n \leq 15$ ).

### Output

In ra kết quả mỗi test trên một dòng là số lớn nhất tìm được.

### Example

#### Input:

1

4 1234

#### Output:

33222

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH LŨY THỪA

### Bài làm tốt nhất

Cho ba số nguyên dương  $x, y, p$ . Nhiệm vụ của bạn là tính  $(x^y) \% p$ . Ví dụ với  $x = 2, y = 3, p = 5$  thì  $(2^3) \% 5 = 3$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ ba  $x, y, p$  được viết cách nhau một vài khoảng trống.
- $T, x, y, p$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 1 \leq x, y \leq 10^6; 1 \leq p \leq 10^9 + 7$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
2 3 5	1
3 2 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẢO TỪ

### Bài làm tốt nhất

Cho xâu ký tự S. Nhiệm vụ của bạn là đảo ngược các từ trong S. Ví dụ S = “I like this program very much”, ta nhận được kết quả là “much very program this like I”.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq \text{Length}(S) \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	
I like this program very much	much very program this like I
much very program this like I	I like this program very much

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SAO CHÉP TỆP TIN

### Bài làm tốt nhất

Cho tệp dữ liệu đầu vào có tên: PTIT.in đặt cùng thư mục với tệp mã nguồn.

Nhiệm vụ của bạn là sao chép nội trong tệp PTIT.in tới tệp PTIT.out

Ví dụ:

PTIT.in	PTIT.out
Ngon ngu lap trinh C++	Ngon ngu lap trinh C++

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CẤU TRÚC SINH VIÊN

### Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float).

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **N20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

### Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 Tab.

### Ví dụ

Input	Output
Nguyen Van A D20CQCN01-N 2/2/2000 2.5	N20DCCN001 Nguyen Van A D20CQCN01-N 02/02/2000 2.50

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct SinhVien a;
    nhapThongTinSV(a);
    inThongTinSV(a);
}
```



```
    return 0;  
}
```

## **SỐ TĂNG GIẢM**

### **Bài làm tốt nhất**

Một số được gọi là số tăng giảm nếu số đó có các chữ số thỏa mãn hoặc tăng dần, hoặc giảm dần từ trái qua phải. Hãy đếm các số nguyên tố là số tăng giảm với **số chữ số cho trước**.

**Dữ liệu vào:** Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng số chữ số tương ứng cần kiểm tra (lớn hơn 1 và nhỏ hơn 10)

**Kết quả:** Ghi ra số lượng các số thỏa mãn điều kiện.

Input	Output
2	2 0
2	5 0
4	

Giới hạn thời gian: 4s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ NGUYÊN TỐ NHỎ NHẤT

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là in ra ước số nguyên tố nhỏ nhất của các số từ 1 đến  $N$ . Trong bài toán này, ta coi ước số nguyên tố nhỏ nhất của 1 là 1. Ước số nguyên tố nhỏ nhất của các số chẵn là 2. Ước số nguyên tố nhỏ nhất của các số nguyên tố là chính nó.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số  $N$  được ghi trên một dòng.
- $T, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10000$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 2 5 2
6	1 2 3 2 5 2 7 2 3 2
10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ SPHENIC

### Bài làm tốt nhất

Số nguyên dương  $N$  được gọi là số Sphenic nếu  $N$  được phân tích duy nhất dưới dạng tích của ba số khác nhau. Ví dụ  $N=30$  là số Sphenic vì  $30 = 2 \times 3 \times 5$ ;  $N = 60$  không phải số Sphenic vì  $60 = 2 \times 2 \times 3 \times 5$ . Cho số tự nhiên  $N$ , nhiệm vụ của bạn là kiểm tra xem  $N$  có phải số Sphenic hay không?

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương  $N$ .
- $T, N$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10000$ .

Output:

- Đưa ra 1 hoặc 0 tương ứng với  $N$  là số Sphenic hoặc không của mỗi test theo từng dòng.

Input:	Output:
2	1
30	0
60	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ CHIA HẾT CHO 2

### Bài làm tốt nhất

Cho số nguyên dương N.

Nhiệm vụ của bạn là hãy xác định xem có bao nhiêu ước số của N chia hết cho 2?

#### Input:

Dòng đầu tiên là số lượng bộ test T ( $T \leq 100$ ).

Mỗi bộ test gồm một số nguyên N ( $1 \leq N \leq 10^9$ )

#### Output:

Với mỗi test, in ra đáp án tìm được trên một dòng.

#### Ví dụ:

Input:	Output:
2	0
9	3
8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CHIA HẾT CHO A VÀ B

### Bài làm tốt nhất

Cho bốn số M, N, A, B. Nhiệm vụ của bạn là hãy đếm tất cả các số trong khoảng [M, N] chia hết cho các số A, B.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 4 số M, N, A, B.
- T, M, N, A, B thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq M, N \leq 10^5$ ;  $1 \leq A, B \leq 500$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 5 11 4 6 3 1000 5 9	2 289

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ NGUYÊN TỐ NHỎ NHẤT

### Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đưa ra ước số nguyên tố nhỏ nhất của các số từ 1 đến N.

Ví dụ n=10, ta có được kết quả : 1 2 3 2 5 2 7 2 3 2.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N.
- T, N thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^9$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 2 5
5	1 2 3 2 5 2 7 2 3 2
10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **ĐẾM SỐ CÓ BA ƯỚC SỐ**

### **Bài làm tốt nhất**

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy đếm tất cả các số có đúng ba ước số. Ví dụ  $n=100$ , ta có các số 4.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số  $N$ .
- $T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100; 1 \leq N \leq 10^{12}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
50	6
200	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## KIỂM TRA SỐ FIBONACCI

### Bài làm tốt nhất

Cho số nguyên dương  $n$ . Hãy kiểm tra xem  $n$  có phải là số Fibonacci hay không?

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương  $n$ .
- $T, n$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 0 \leq n \leq 10^{18}$ .

Output:

- Đưa ra “YES” hoặc “NO” tương ứng với  $n$  là số Fibonacci hoặc không phải số Fibonacci của mỗi test theo từng dòng.

Input	Output
2	YES
8	NO
15	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## NGUYÊN TỐ CÙNG NHAU

### Bài làm tốt nhất

Juggernaut được cô giáo Disruptor dạy toán, cô giáo định nghĩa một hàm  $f(x)$  như sau:

Với  $t$  là số lượng các số tự nhiên  $k$  ( $1 \leq k \leq x$ ) thỏa mãn nguyên tố cùng nhau với  $x$ , nếu  $t$  là nguyên tố thì  $f(x) = 1$ , ngược lại  $f(x) = 0$ .

Disruptor cho Juggernaut một số nguyên dương  $x$ , yêu cầu anh cho biết giá trị của hàm  $f(x)$ , nếu trả lời sai thì Jug sẽ bị cô trả về nhà, Jug không muốn về nhà, các bạn hãy giúp Jug giải bài toán này.

### Input

Dòng đầu tiên chứa số bộ test  $T$  ( $T \leq 10$ ).

Mỗi test gồm một dòng chứa số  $x$  ( $1 \leq x \leq 10^5$ ).

### Output

In ra kết quả mỗi test trên một dòng là giá trị của hàm  $f(x)$ .

### Example

#### Input:

2

2

3

#### Output:

0

1

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## CHIA HẾT CHO BA SỐ NGUYÊN

### Bài làm tốt nhất

Cho bốn số nguyên dương  $X, Y, Z$  và  $N$ . Hãy tìm số nguyên dương nhỏ nhất có  $N$  chữ số chia hết đồng thời cho  $X, Y, Z$ . Ví dụ với  $X = 2, Y = 3, Z = 5, N = 4$  ta tìm được số nguyên dương nhỏ nhất có 4 chữ số là 1020 chia hết cho cả 2, 3, 5.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$  ( $T \leq 100$ ).
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là bộ bốn số  $X, Y, Z, N$ . Các số  $X, Y, Z, N$  thỏa mãn ràng buộc dưới đây:

$$1 \leq X, Y, Z \leq 10^5; N \leq 18.$$

Output:

- Đưa ra theo từng dòng kết quả mỗi test là số nguyên nhỏ nhất có  $N$  chữ số chia hết đồng thời cho  $X, Y, Z$ . Trong trường hợp không có số nguyên  $N$  chữ số thỏa mãn yêu cầu bài toán đưa ra giá trị -1.

Input	Output
3	1020
2 3 5 4	120
4 5 6 3	-1
3 5 7 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ LẶP LẠI

### Bài làm tốt nhất

Cho ba số  $a, x, y$ . Nhiệm vụ của bạn là tìm ước số chung lớn nhất của hai số  $P$  và  $Q$ , trong đó  $P$  lặp lại  $x$  lần số  $a$  và  $Q$  lặp lại  $y$  lần số  $a$ . Ví dụ  $a=2, x=3, y=2$  thì  $P=222, Q=22$ .

Input:

- Dòng đầu tiên đưa vào  $T$  là số lượng bộ test.
- $T$  dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là bộ ba số  $a, x, y$  phân biệt nhau bởi một vài khoảng trống.
- Các số  $T, a, x, y$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq a, x, y \leq 10^{18}$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	2
2 2 3	123
123 5 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẾM SỐ NGUYÊN TỐ

### Bài làm tốt nhất

Cho một câu hỏi  $Q$  là bộ đôi hai số  $L$  và  $R$ . Nhiệm vụ của bạn là xác định xem có bao nhiêu số nguyên tố trong khoảng  $[L, R]$ . Ví dụ với  $Q = [1, 10]$  ta có câu trả lời là 4 vì có  $\{2, 3, 5, 7\}$  là các số nguyên tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một bộ đôi  $L, R$ . các số được viết cách nhau một vài khoảng trống.
- $T, L, R$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq L \leq R \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
1 10	2
5 10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## HIỆU LỚN NHẤT CỦA CẶP PHẦN TỬ ĐÚNG THỨ TỰ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên. Hãy tìm hiệu lớn nhất của bất kể hai phần tử nào của mảng dãy con thỏa mãn ràng buộc số lớn hơn xuất hiện sau số nhỏ hơn. Nếu không tìm được cặp phần tử của mảng hãy đưa ra -1. Ví dụ với mảng  $A[] = \{2, 3, 10, 6, 4, 8, 1\}$  ta nhận được kết quả là  $8 = 10 - 2$ . Với mảng  $A[] = \{7, 9, 5, 6, 3, 2\}$  ta có kết quả là  $2 = 9 - 7$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $1 \leq A[i] \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	8
7	-1
2 3 10 6 4 8 1	
3	
3 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH GIÁ TRỊ ĐA THỨC

### Bài làm tốt nhất

Tính toán giá trị đa thức  $P(n, x) = a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \dots + a_0$ .

Kết quả có thể rất lớn nên hãy chia dư cho  $10^9 + 7$

#### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa vào hai số n, x; dòng tiếp theo đưa vào n số  $a_{n-1}, a_{n-2}, \dots, a_0$  là hệ số của đa thức P. Các số được viết cách nhau một vài khoảng trống.
- T, n, x,  $P[i]$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $0 \leq n \leq 2000$ ;  $0 \leq x, P[i] \leq 1000$ .
- **Chú ý:** Các hệ số của đa thức P được viết theo thứ tự từ bậc 0 đến bậc n-1

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 4 2  1 2 0 4	20

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## KIỂM TRA DÃY FIBONACCI

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên không âm. Hãy tìm dãy con lớn nhất chỉ toàn các số Fibonacci. Số 0 được coi là số Fibonacci đầu tiên.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng tiếp theo đưa vào  $n$  số các phần tử của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 100$ ;  $1 \leq A[i] \leq 1000$ .

Output:

- Đưa ra dãy con lớn nhất bao gồm các số Fibonacci của mỗi test theo từng dòng.

Input	Output
2	1 3 13
7	0 2 8 5 2 1 13
1 4 3 9 10 13 7	
9	
0 2 8 5 2 1 4 13 23	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## **DÃY MOUNTAIN**

### **Bài làm tốt nhất**

Cho mảng  $A[]$  gồm  $n$  phần tử và một câu hỏi  $Q$ . Mỗi câu hỏi  $Q$  là bộ đôi hai số  $L$  và  $R$ . Nhiệm vụ của bạn là xác định xem dãy con của  $A[]$  trong khoảng  $[L, R]$  có tạo nên một dãy Mountain hay không? Dãy Mountain là dãy được chia thành hai phần: phần thứ nhất tăng phần thứ hai giảm. Ví dụ với mảng  $A[] = \{2, 3, 2, 4, 4, 6, 3, 2\}$ , các câu hỏi  $Q$ :  $[0, 2]$ ,  $[2, 7]$ ,  $[2, 3]$ ,  $[1, 3]$  ta sẽ có các câu trả lời: Yes, Yes, Yes, No tương ứng với các dãy Mountain  $[2, 3, 2]$ ,  $[2, 4, 4, 6, 3, 2]$ ,  $[2, 4]$ ,  $[3, 2, 4]$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào  $n$ ,  $L$ ,  $R$  là số phần tử của mảng  $A[]$  và câu hỏi  $Q$ ; phần tiếp theo đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, L, R, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq L \leq R \leq n$ ,  $\leq 10^4$ ;  $1 \leq A[i] \leq 10^3$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	Yes
8	No
2 3 2 4 4 6 3 2	
0 2	
8	
2 3 2 4 4 6 3 2	
1 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## PHẦN TỬ NHỎ NHẤT THỨ K CỦA MA TRẬN

### Bài làm tốt nhất

Cho ma trận vuông  $A[][]$  cấp  $n$ . Các phần tử của ma trận  $A[][]$  đã được sắp xếp theo hàng, cột. Hãy tìm phần tử nhỏ nhất thứ  $k$  của ma trận. Ví dụ với ma trận cấp 4 dưới đây sẽ cho ta số nhỏ nhất thứ 3 là 20, số nhỏ nhất thứ 7 là 30.

10 20 30 40

15 25 35 45

24 29 37 48

32 33 39 50

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là  $n$  và  $k$ ; phần thứ hai là  $n^2$  các phần tử của ma trận vuông  $A[][]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 100$ ;  $1 \leq k, A[i][j] \leq 10^4$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	30
4 7	
10 20 30 40	
15 25 35 45	
24 29 37 48	
32 33 39 50	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## QUAY MA TRẬN

### Bài làm tốt nhất

Cho ma trận  $A[][]$  gồm các số nguyên dương. Nhiệm vụ của bạn là quay ma trận theo chiều kim đồng hồ. Ví dụ về quay theo chiều kim đồng hồ ma trận  $A[][]$  dưới đây.

1	2	3	4	1	2
4	5	6	7	5	3
7	8	9	8	9	6

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n, m$  tương ứng với số hàng, số cột của ma trận  $A[]$ ; dòng tiếp theo đưa vào  $n \times m$  số  $A[i][j]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, m \leq 100$ ;  $1 \leq A[i][j] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5 1 6 2
2 2	
1 2 5 6	4 1 2 7 5 3 8 9 6
3 3	
1 2 3 4 5 6 7 8 9	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẾM PHẦN TỬ GIỐNG NHAU

### Bài làm tốt nhất

Cho ma trận vuông  $A[][]$  cấp  $N$ . Nhiệm vụ của bạn là đưa ra số các phần tử giống nhau ở tất cả các hàng. Ví dụ với ma trận  $A[][]$  dưới đây sẽ cho ta kết quả là 2 tương ứng với số 2, 3 xuất hiện ở tất cả các hàng.

2	1	4	3
1	2	3	2
3	6	2	3
5	2	5	3

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $N$  là cấp của ma trận  $A[][]$ ; dòng tiếp theo đưa vào  $N \times N$  số  $A[i][j]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, N, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 100$ ;  $1 \leq A[i][j] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 4 2 1 4 3 1 2 3 2 3 6 2 3 5 2 5 3	2

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Giới hạn bộ nhớ: 65536 Kb

## BIẾN ĐỔI MA TRẬN

### Bài làm tốt nhất

Cho ma trận vuông  $A[N][N]$ . Tìm số phép biến đổi ít nhất để tổng theo các hàng, các cột của ma trận đều bằng nhau. Biết mỗi phép biến đổi bạn chỉ được phép tăng một phần tử bất kỳ của ma trận lên 1 đơn vị. Ví dụ với ma trận

$$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \rightarrow \begin{vmatrix} 4 & 3 \\ 3 & 4 \end{vmatrix}$$

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $N$  là cấp của ma trận  $A[N][N]$ ; dòng tiếp theo đưa vào  $N \times N$  số  $A[i][j]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, N, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 100$ ;  $1 \leq A[i][j] \leq 150$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
2	6
1 2 3 4	
3	
1 2 3 4 2 3 3 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## BIẾN ĐỔI DÃY SỐ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Hãy biến mảng  $A[]$  thành một mảng đối xứng sao cho phép thay thế  $A[i] = \text{Merge}(A[i], A[i+1])$  được thực hiện ít nhất. Trong đó,  $\text{Merge}(A[i], A[i+1]) = A[i] + A[i+1]$ . Ví dụ với  $A[] = \{3, 2, 3, 3, 5\}$  ta chỉ cần thực hiện 1 phép  $\text{Merge}(A[0], A[1])$  để trở thành mảng  $A[] = \{5, 3, 3, 5\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $1 \leq A[i] \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5	3
3 2 3 3 5	
4	
5 3 3 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## DÃY SỐ NHỊ PHÂN

### Bài làm tốt nhất

Cho mảng các số nhị phân  $A1[]$  và  $A2[]$  gồm  $n$  0, 1. Hãy tìm khoảng chung dài nhất thỏa mãn:  $j \geq i$  và  $\text{span}(i, j) = A1[i] + A1[i+1] + \dots + A1[j] = A2[i] + A2[i+1] + \dots + A2[j]$ . Ví dụ với  $A1[] = \{0, 1, 0, 0, 0, 0\}$ ,  $A2[] = \{1, 0, 1, 0, 0, 1\}$  ta có kết quả là 4 tương ứng với  $A1[1] + A1[2] + A1[3] + A1[4] = A2[1] + A2[2] + A2[3] + A2[4] = 1$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên là số phần tử của mảng  $n$ ; dòng tiếp theo là  $n$  số  $A1[i]$  của mảng  $A1[]$ ; dòng tiếp theo là  $n$  số  $A2[i]$  của mảng  $A2[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 6 0 1 0 0 0 0 1 0 1 0 0 1	4

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## MA TRẬN XOẮN ỐC - 4

### Bài làm tốt nhất

Cho ma trận vuông A cỡ  $N \times N$  chỉ bao gồm các số nguyên dương không quá 1000. Hãy sắp đặt các giá trị trong ma trận A sao cho các số được điền lần lượt theo kiểu xoắn ốc tăng dần, theo chiều kim đồng hồ.

### Input

Dòng đầu ghi số N ( $2 < N < 20$ ).

N dòng tiếp theo ghi ma trận A, các giá trị nguyên dương và không quá 1000.

### Output

Ghi ra ma trận kết quả

### Ví dụ

Input	Output
3	1 3 4
3 6 1	9 12 5
8 7 9	8 7 6
4 12 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CỬA SỐ TRƯỢT

### Bài làm tốt nhất

Cho ma trận vuông A cỡ  $N \times N$ . Một ma trận vuông B nhỏ hơn cỡ  $M \times M$  có thể dùng làm “cửa sổ trượt” trên ma trận A nếu M là ước số của N.

Hãy thực hiện tính tích chập của ma trận B với từng “khung cửa sổ” tương ứng trên ma trận A. Tích chập được hiểu là tính giá trị tích từng vị trí tương ứng trên 2 ma trận kích thước bằng nhau.

Xem ví dụ để hiểu rõ hơn.

### Input

Dòng đầu ghi số N ( $3 < N < 100$ ). Tiếp theo là N dòng ghi ma trận A. Các giá trị đều nguyên dương và không quá 1000.

Tiếp theo là một dòng ghi số M ( $1 < M < 10$ ). Tiếp theo là M dòng ghi ma trận B. Các giá trị lớn hơn hoặc bằng 0 và không quá 20.

Dữ liệu vào đảm bảo M là ước số của N.

### Output

Ghi ra N dòng mô tả ma trận kết quả.

### Ví dụ

Input	Output
4	1 0 3 0
1 2 3 4	0 12 0 16
5 6 7 8	9 0 11 0
9 10 11 12	0 28 0 32
13 14 15 16	
2	
1 0	
0 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐIỂM CÂN BẰNG

### Bài làm tốt nhất

Cho dãy số  $A[]$  gồm có  $N$  phần tử nguyên dương. Phần tử thứ  $i$  được gọi là điểm cân bằng của dãy số nếu như tổng các số bên trái bằng tổng các số bên phải của nó.

Nhiệm vụ của bạn là điểm cân bằng đầu tiên của dãy  $A[]$  cho trước. Nếu không có đáp án, in ra -1.

### Input:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test gồm số nguyên  $N$  ( $1 \leq N \leq 100\,000$ ), số lượng phần tử trong dãy số ban đầu.

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $-1000 \leq A[i] \leq 1000$ ).

### Output:

Với mỗi test, in ra trên một dòng vị trí của điểm cân bằng tìm được.

### Ví dụ:

Input:	Output
2	4
7	-1
-7 1 5 2 -4 3 0	
5	
1 2 3 4 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## HIỆU CỦA HAI TẬP TỪ

### Bài làm tốt nhất

Cho hai xâu ký tự S1 và S2. Hãy viết chương trình tìm các từ chỉ xuất hiện trong S1 mà không xuất hiện trong S2. Chú ý: mỗi từ chỉ liệt kê 1 lần.

**Dữ liệu vào:** Dòng 1 ghi số bộ test. Mỗi bộ test gồm 2 dòng, mỗi dòng ghi một xâu ký tự độ dài không quá 200, chỉ bao gồm các ký tự viết thường và các khoảng trống.

**Kết quả:** Với mỗi bộ test ghi ra các từ có trong S1 mà không có trong S2. Các từ được ghi theo thứ tự từ điển.

Ví dụ:

Input	Output
2	abcd
abc ab ab ab abcd	aaa ab abc zzz
ab abc	
aaa xyz ab zzz abc dd dd abc	
xyz dd ttt sas cdc	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TỔNG LỚN NHẤT VÀ NHỎ NHẤT

### Bài làm tốt nhất

Cho hai số nguyên dương  $X1$ ,  $X2$ . Ta chỉ được phép thay đổi chữ số 5 thành 6 và ngược lại chữ số 6 thành chữ số 5 của các số  $X1$  và  $X2$ . Hãy đưa ra tổng nhỏ nhất và tổng lớn nhất các số  $X1$  và  $X2$  được tạo ra theo nguyên tắc kể trên.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là cặp các số  $X1$ ,  $X2$ .
- $T$ ,  $X1$ ,  $X2$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq X1, X2 \leq 10^{18}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1100 1312
645 666	10010 11132
5466 4555	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP ĐẶT XÂU KÝ TỰ - 1

### Bài làm tốt nhất

Cho chuỗi ký tự S bao gồm các ký tự 'a', ..., 'z'. Các ký tự trong S có thể lặp lại. Nhiệm vụ của bạn sắp đặt lại các ký tự trong S sao cho các ký tự kề nhau đều khác nhau.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một chuỗi ký tự S.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq \text{Length}(S) \leq 10^3$ .

Output:

- Đưa ra 1 hoặc 0 nếu có thể hoặc không thể sắp đặt lại các ký tự trong chuỗi S thỏa mãn yêu cầu bài toán.

Input:	Output:
3	1
geeksforgeeks	1
bbbabaaacd	0
bbbbbb	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐỔI CHỖ CHỮ SỐ

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Bạn chỉ được phép sử dụng nhiều nhất một phép đổi chỗ giữa 2 chữ số để nhận được số lớn nhất nhỏ hơn  $N$ . Ví dụ với số  $N=12435$ , sử dụng một phép đổi chỗ ta nhận được số lớn nhất nhỏ hơn  $N$  là 12345. Mặc dù  $12354 > 12345$  nhưng ta không thể tạo ra số 12345 với chỉ một phép hoán vị. Với số  $N=12345$  ta không có phép đổi chỗ.

Input:

- Dòng đầu tiên đưa vào  $T$  là số lượng bộ test.
- $T$  dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là một xâu ký tự số không có ký tự '0' đầu tiên.
- Các số  $T, N$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq \text{length}(N) \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng, trong đó -1 được xem là test không có phép đổi chỗ.

Input	Output
2	12345
12435	-1
12345	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## SỐ MAY MẮN - 2

### Bài làm tốt nhất

Một số nguyên không âm  $n$  được gọi là số may mắn nếu tổng các chữ của  $n$  bằng 9 hoặc tổng các chữ số của  $n$  là số may mắn. Ví dụ các số 9, 108, 279 là các số may mắn, còn các số 19, 289 không phải là số may mắn.

**Yêu cầu:** Cho số nguyên không âm  $n$ , hãy kiểm tra xem  $n$  có phải là số may mắn hay không?

### Input

Dữ liệu vào gồm nhiều bộ dữ liệu tương ứng với nhiều test. Dòng đầu tiên chứa một số nguyên dương không vượt quá 100 là số lượng các bộ dữ liệu. Các dòng tiếp theo chứa các bộ dữ liệu.

Mỗi bộ dữ liệu gồm một dòng duy nhất chứa một số nguyên không âm  $n$  ( $n \leq 10^{100}$ ).

### Output

Với mỗi bộ dữ liệu, ghi ra trên một dòng câu trả lời, ghi số 1 nếu  $n$  là số may mắn, ghi số 0 trong trường hợp ngược lại.

### Example

**Input:**

3  
888  
666  
289

**Output:**

0  
1  
0

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **BIỂN SỐ ĐẸP**

### **Bài làm tốt nhất**

Biển số xe máy được quy định gồm các thành phần:

- Hai chữ số đầu là mã quản lý theo tỉnh – thành phố (ví dụ mã của Hà Nội là 29 đến 31)
- Tiếp theo là dấu gạch ngang, sau đó là một chữ cái in hoa (từ A đến Z) và một chữ số. Cặp chữ cái và chữ số này được cấp theo khu vực quận – huyện.
- Tiếp theo là một dãy 5 số gồm 2 cụm: 3 chữ số đầu và hai chữ số sau, phân tách bởi dấu chấm.

Thông thường, người ta chỉ quan tâm đến 5 chữ số cuối. Giả sử các trường hợp sau được coi là đẹp:

- Cả 5 chữ số được sắp theo thứ tự tăng chặt từ trái qua phải.
- Cả 5 chữ số đều bằng nhau
- Ba chữ số đầu bằng nhau và hai chữ số cuối bằng nhau
- Cả 5 chữ số đều là 6 và/hoặc 8 (số lộc phát).

Theo quy tắc trên, các biển số sau được coi là đẹp:

- 29-T1 123.79
- 29-T1 555.55
- 29-T1 222.33
- 29-T1 686.88

Và các biển số sau không đẹp:

- 29-T1 123.33
- 29-T1 555.54
- 29-T1 606.88

Viết chương trình kiểm tra xem các biển số xe có đẹp hay không.

### **Dữ liệu vào**

- Dòng đầu ghi số bộ test, không quá 50
- Mỗi bộ test là một biển số. Dữ liệu vào đảm bảo biển số được viết đúng quy định.

### **Kết quả**

- In ra kết quả kiểm tra với từng bộ test

### Ví dụ

Input	Output
7	YES
29T1-123.45	YES
29T1-555.55	YES
29T1-222.33	YES
29T1-686.88	NO
29T1-123.33	NO
29T1-555.54	NO
29T1-606.88	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ ĐẦY ĐỦ

### Bài làm tốt nhất

Cho một số nguyên dương lớn có nhiều hơn 20 chữ số nhưng không quá 1000 chữ số. Hãy kiểm tra xem số đó có đầy đủ tất cả các chữ số từ 0 đến 9 hay không.

### Dữ liệu vào

- Dòng đầu ghi số bộ test, không quá 10
- Mỗi bộ test là một dãy ký tự có độ dài không quá 1000, không có khoảng trống

### Kết quả

- Nếu dữ liệu vào không phải là một số nguyên hợp lệ (có ký tự không phải số hoặc bắt đầu bằng chữ số 0) thì in ra INVALID
- Nếu dữ liệu vào thỏa mãn đầy đủ thì in ra YES, nếu không in ra NO

### Ví dụ

Input	Output
3	INVALID
01234aa32432432432534545b987978	YES
1234444444444444566666666667890	NO
324562783924723543243243242354354354333234324	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## HIỆU HAI SỐ NGUYÊN LỚN

### Bài làm tốt nhất

Cho hai số rất lớn X và Y được biểu diễn như hai chuỗi ký tự. Nhiệm vụ của bạn là tìm  $|X-Y|$ ?

#### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa chuỗi X; dòng tiếp theo đưa vào chuỗi Y.
- T, X, Y thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $0 \leq \text{length}(X), \text{length}(Y) \leq 10^3$ .

#### Output:

- Đưa ra số kết quả mỗi test theo từng dòng.

#### Ví dụ:

Input:	Output:
2	11999
978	
12977	0999900
100	
1000000	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TỔNG HAI SỐ NGUYÊN LỚN

### Bài làm tốt nhất

Cho hai số rất lớn X và Y được biểu diễn như hai chuỗi ký tự. Nhiệm vụ của bạn là tìm  $X+Y$ ?

#### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa chuỗi X; dòng tiếp theo đưa vào chuỗi Y.
- T, X, Y thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $0 \leq \text{length}(X), \text{length}(Y) \leq 10^3$ .

#### Output:

- Đưa ra số kết quả mỗi test theo từng dòng.

#### Ví dụ:

Input:	Output:
1	198123
12	
198111	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## PHÉP CHIA DƯ CỦA SỐ NGUYÊN LỚN

### Bài làm tốt nhất

Cho số nguyên dương  $N$  rất lớn được biểu diễn như một chuỗi và số  $M$ . Hãy tìm  $K = N \% M$ . Ví dụ  $N=123456789873123456778976$ ,  $M = 100$  thì  $K=76$ .

#### Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp mỗi dòng đưa vào các test. Mỗi test là bộ đôi  $N, M$  được viết trên hai dòng khác nhau.
- $T, N, M$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $0 \leq \text{length}(N) \leq 1000$ ;  $2 \leq M \leq 10^{12}$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 12345 10	5

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## PHÉP CHIA DƯ CỦA LŨY THỪA SỐ NGUYÊN LỚN

### Bài làm tốt nhất

Cho số nguyên dương  $a, b, M$ , trong đó  $a$  là số rất lớn được biểu diễn như một xâu ký tự số. Hãy tìm  $K = (a^b) \% M$ . Ví dụ  $a = 3, b=2, M = 4$  thì  $K = (3^2)\%4 = 1$

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ ba  $a, b, M$  được viết trên một dòng.
- $T, a, b, M$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 0 \leq \text{length}(a) \leq 1000; 2 \leq b, M \leq 10^{12}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 3 2 4	1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## TÍNH TỔNG CÁC SỐ TRONG XÂU

### Bài làm tốt nhất

Cho chuỗi ký tự S bao gồm các ký tự 'a', ..., 'z' và các chữ số. Nhiệm vụ của bạn là hãy tính tổng các số có mặt trong chuỗi.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một chuỗi ký tự S.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq \text{Length}(S) \leq 10^5$ .
- Input đảm bảo đáp án không vượt quá  $10^9$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
4	24
1abc23	4
geeks4geeks	100
1abc2x30yz67	123
123abc	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÌM SỐ LỚN NHẤT TRONG XÂU

### Bài làm tốt nhất

Cho xâu ký tự S bao gồm các ký tự 'a', ..., 'z' và các chữ số. Nhiệm vụ của bạn là hãy tìm số lớn nhất có mặt trong xâu.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq \text{Length}(S) \leq 10^5$ .
- Input đảm bảo đáp số không vượt quá  $10^9$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	564
100klh564abc365bg	9
abvhd9sdnkjdfs	0
abchsd0sdhs	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẾM XÂU CON

### Bài làm tốt nhất

Cho chuỗi ký tự  $S$  và số  $k$ . Nhiệm vụ của bạn là đếm số chuỗi con của  $S$  có đúng  $k$  ký tự khác nhau. Các chuỗi con không nhất thiết phải khác nhau. Ví dụ với chuỗi  $S = \text{"abc"}$  và  $k = 2$ , ta có kết quả là 2 bao gồm các chuỗi con: "ab", "bc".

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một chuỗi ký tự  $S$  và số  $k$  được viết trên một dòng.
- $T, S$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq k \leq 26$ ;  $1 \leq \text{Length}(S) \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
abc 2	3
aba 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÁCH CHỮ SỐ

### Bài làm tốt nhất

Cho xâu ký tự S bao gồm các ký tự 'A',..., 'Z' và các chữ số '0',..., '9'. Nhiệm vụ của bạn in các ký tự từ 'A',..., 'Z' trong S theo thứ tự alphabet và nối với tổng các chữ số trong S ở cuối cùng. Ví dụ S="ACCBA10D2EW30" ta nhận được kết quả: "AABCCDEW6".

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq \text{Length}(S) \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	ABCEW5
AC2BEW3	AABCCDEW6
ACCBA10D2EW30	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CHUẨN HÓA HỌ TÊN

### Bài làm tốt nhất

Khi viết tên người Việt Nam sang dạng chuẩn Tiếng Anh không dấu, người ta có thể lựa chọn 1 trong 2 cách viết:

- Cách 1: tên có 1 từ.

Ví dụ: Nguyen Manh Son được viết thành Son Nguyen Manh

- Cách 2: họ có 1 từ.

Ví dụ: Nguyen Manh Son được viết thành Manh Son Nguyen

Hãy viết chương trình chuẩn hóa xâu ký tự họ tên theo một trong hai dạng trên.

### Input

Dòng đầu ghi số bộ test.

Mỗi bộ test gồm 2 dòng. Dòng đầu ghi số 1 hoặc số 2 cho biết cách chuẩn hóa được chọn. Dòng thứ 2 ghi xâu họ tên có độ dài không quá 60 ký tự.

### Output

Ghi ra danh sách các xâu họ tên đã chuẩn hóa.

Chú ý: có thể phải loại bỏ khoảng trống, chuyển đổi chữ hoa chữ thường cho đúng với cách viết họ tên chuẩn trước khi áp dụng một trong hai cách viết nói trên.

### Ví dụ

Input	Output
2	Nam Nguyen Van
1	Le Nhat Thong Chi Hoang
NguyenN      vAn      Nam	
2	
hoAng   le Nhat THONG chi	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐỊA CHỈ EMAIL - 2

### Bài làm tốt nhất

Địa chỉ email của các cán bộ, giảng viên PTIT được tạo ra bằng cách viết đầy đủ tên và ghép với các chữ cái đầu của họ và tên đệm. Nếu có nhiều người cùng email thì từ người thứ 2 sẽ thêm số thứ tự vào email đó.

Cho trước các xâu họ tên (có thể không chuẩn). Hãy tạo ra các địa email tương ứng.

#### Input:

- Dòng 1 ghi số N là xâu họ tên trong danh sách
- N dòng tiếp theo ghi lần lượt các xâu họ tên (không quá 50 ký tự)

**Output:** Ghi ra các email được tạo ra.

#### Ví dụ:

Input	Output
4	vinhnq@ptit.edu.vn
nGUYEn quaNG vInH	huongttt@ptit.edu.vn
tRan thi THU huOnG	vinhnq2@ptit.edu.vn
nGO quoC VINH	anhlt@ptit.edu.vn
lE tuAn aNH	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## ĐIỆN THOẠI CỤC GẠCH

### Bài làm tốt nhất

Một thời không quá xa, điện thoại di động với chỉ các tính năng nghe, gọi, nhắn tin vẫn còn chiếm đại đa số thiết bị cầm tay tại Việt Nam. Khi nhắn tin, người nhắn sẽ bấm các phím số một đến bốn lần liên tiếp tương ứng với ký tự đi kèm ghi trên đó.

Cụ thể: các số và chữ cái tương ứng gồm:

2: ABC, 3: DEF, 4: GHI, 5: JKL

6: MNO, 7: PQRS, 8: TUV, 9: WXYZ

Cho trước dãy ký tự mô tả tin nhắn (không tính các ký tự khác ngoài danh sách nêu trên). Hãy kiểm tra xem dãy số được nhấn ứng với dãy ký tự đó có phải số thuận nghịch hay không (chỉ xét tương ứng giữa số và ký tự, không tính số đó được nhấn bao nhiêu lần, ví dụ tất cả A,B,C,a,b,c đều chỉ là một chữ số 2).

### Input

Dòng đầu tiên là số bộ test, không quá 1000.

Mỗi test là dãy ký tự mô tả tin nhắn.

### Output

Ghi ra kết quả kiểm tra, YES nếu dãy số là thuận nghịch, NO nếu ngược lại.

### Ví dụ

Input	Output
2	YES
BOHIMA	NO
IamACoder	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## XỬ LÝ VĂN BẢN

### Bài làm tốt nhất

Cho dữ liệu vào là luồng văn bản bất kỳ, gồm các ký tự viết hoa, viết thường, các ký tự số và các dấu câu, không có các ký tự đặc biệt khác. Người ta muốn tách văn bản thành các câu. Mỗi câu in trên một dòng.

Một câu được định nghĩa là dãy ký tự có *ít nhất 1 ký tự chữ cái hoặc chữ số*, không chứa các dấu ngắt câu gồm: dấu chấm (.), dấu chấm hỏi (?) và dấu chấm cảm (!).

Dấu phẩy (,) và dấu hai chấm (:) không được coi là dấu ngắt câu.

Nhiệm vụ của bạn là in ra mỗi câu trên một dòng, ký tự đầu câu viết hoa, các ký tự khác viết thường, các từ cách nhau đúng một khoảng trống. Không có khoảng trống ở đầu và cuối câu, và không in ra các dấu ngắt câu.

### Input

Gồm một luồng văn bản không quá 100 dòng, mỗi dòng không quá 200 ký tự.

### Output

In ra các câu, mỗi câu trên một dòng theo quy tắc đã cho.

### Ví dụ

Input
ky thi LAP TRINH ICPC PTIT bat dau to chuc tu nam 2010. nhu vay, nam nay la tron 10 nam!
vay CO PHAI NAM NAY LA KY THI LAN THU 10? khong phai! nam nay la KY THI LAN THU 11.
Output

Ky thi lap trinh icpc ptit bat dau to chuc tu nam 2010

Nhu vay, nam nay la tron 10 nam

Vay co phai nam nay la ky thi lan thu 10

Khong phai

Nam nay la ky thi lan thu 11

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP LẠI DẪY CON

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Hãy tìm dãy con liên tục của mảng  $A[R], \dots, A[L]$  sao cho khi sắp xếp lại dãy con ta nhận được một mảng được sắp xếp. Ví dụ với  $A[] = \{10, 12, 20, 30, 25, 40, 32, 31, 35, 50, 60\}$  ta chỉ cần sắp xếp lại dãy con từ  $A[4], \dots, A[9]: \{30, 25, 40, 32, 31, 35\}$  để có mảng được sắp.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$  các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^6$ ;  $0 \leq A[i] \leq 10^7$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 9
11	3 6
10 12 20 30 25 40 32 31 35 50 60	
9	
0 1 15 25 6 7 30 40 50	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## HỢP VÀ GIAO CỦA HAI DÃY SỐ -1

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử, mảng  $B[]$  gồm  $m$  phần tử khác nhau. Các phần tử của mảng  $A[]$  và  $B[]$  đã được sắp xếp. Hãy tìm mảng hợp và giao được sắp giữa  $A[]$  và  $B[]$ . Ví dụ với  $A[] = \{1, 3, 4, 5, 7\}$ ,  $B[] = \{2, 3, 5, 6\}$  ta có mảng hợp  $\text{Union} = \{1, 2, 3, 4, 5, 6, 7\}$ , mảng giao  $\text{Intersection} = \{3, 5\}$ . In ra đáp án theo giá trị phần tử từ nhỏ đến lớn.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào  $n, m$  là số phần tử của mảng  $A[]$  và  $B[]$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; dòng tiếp theo là  $m$  số  $B[i]$  của mảng  $B[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A[i], B[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, m, A[i], B[i] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	1 2 3 4 5
5 3	
1 2 3 4 5	1 2 3
1 2 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## HỢP VÀ GIAO CỦA HAI DÃY SỐ - 2

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử, mảng  $B[]$  gồm  $m$  phần tử khác nhau. Các phần tử của mảng  $A[]$  và  $B[]$  chưa được sắp xếp. Hãy tìm mảng hợp và giao được sắp giữa  $A[]$  và  $B[]$ . Ví dụ với  $A[] = \{7, 1, 5, 2, 3, 6\}$ ,  $B[] = \{3, 8, 6, 20, 7\}$  ta có mảng hợp  $\text{Union} = \{1, 2, 3, 5, 6, 7, 8, 20\}$ , mảng giao  $\text{Intersection} = \{3, 6\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào  $n, m$  là số phần tử của mảng  $A[]$  và  $B[]$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; dòng tiếp theo là  $m$  số  $B[i]$  của mảng  $B[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A[i], B[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, m, A[i], B[i] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	1 2 3 5 6 7 8 20
6 5	
7 1 5 2 3 6	3 6
3 8 6 20 7	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP THEO KHOẢNG CÁCH

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử và số  $X$ . Hãy đưa sắp xếp các phần tử của mảng theo trị tuyệt đối của  $|X - A[i]|$ . Ví dụ với  $A[] = \{10, 5, 3, 9, 2\}$  và  $X = 7$  ta đưa ra mảng được sắp xếp theo nguyên tắc kể trên:  $A[] = \{5, 9, 10, 3, 2\}$  vì  $|7-10|=3$ ,  $|7-5|=2$ ,  $|7-3|=4$ ,  $|7-9|=2$ ,  $|7-2|=5$ . Trong trường hợp có nhiều phần tử có giá trị tuyệt đối như nhau, ưu tiên theo thứ tự số xuất hiện trước trong mảng ban đầu.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$  và  $X$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, X$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, X, A[i] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5 9 10 3 2
5 7	5 4 3 2 1
10 5 3 9 2	
5 6	
1 2 3 4 5	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## SỐ NHỎ HƠN K

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương và số  $k$ . Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho các số nhỏ hơn hoặc bằng  $k$  đứng cạnh nhau. Ví dụ với mảng  $A[] = \{2, 1, 5, 6, 3\}$ ,  $k = 3$  ta chỉ cần thực hiện 1 phép đổi chỗ để có mảng  $A[] = \{2, 1, 3, 6, 5\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$  và số  $k$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^7$ ;  $1 \leq A[i], k \leq 10^7$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5 3	2
2 1 5 6 3	
7 5	
2 7 9 5 8 7 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP CHẴN LẺ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Nhiệm vụ của bạn là hãy sắp xếp lại các phần tử của mảng sao cho  $A[i] \geq A[i-1]$  nếu  $i$  chẵn,  $A[i] \leq A[i-1]$  nếu  $i$  lẻ. Ví dụ với mảng  $A[] = \{1, 2, 2, 1\}$  ta được mảng được sắp  $A[] = \{1, 2, 1, 2\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $1 \leq A[i] \leq 10^3$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 1 2
4	1 3 2
1 2 2 1	
3	
1 3 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## SẮP XẾP XEN KẼ - 2

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho nguyên tắc số lớn nhất ở đầu, số nhỏ nhất thứ 2, số lớn thứ nhì ở vị trí tiếp theo, số nhỏ thứ nhì ở vị trí kế tiếp .... Ví dụ với mảng  $A[] = \{1, 7, 3, 5, 9\}$  ta được mảng được sắp  $A[] = \{9, 1, 7, 3, 5\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $1 \leq A[i] \leq 10^3$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7 1 5 2
4	9 1 8 2 7 3
1 5 2 7	
6	
1 3 2 7 9 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## NHÂN ĐÔI CẬP SỐ BẰNG NHAU

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên không âm. Ta gọi phần tử  $A[i]$  là hợp lệ nếu  $A[i] \neq 0$  ngược lại là không hợp lệ. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng theo nguyên tắc nếu số  $A[i+1]$  ( $i=0, \dots, n-2$ ) là số hợp lệ và  $A[i] = A[i+1]$  thì nhân đôi  $A[i]$  và thiết lập  $A[i+1] = 0$ . Sau khi thay đổi, hãy di chuyển các số không hợp lệ vào cuối mảng. Các số hợp lệ phải bảo toàn quan hệ trước sau. Ví dụ với mảng  $A[] = \{2, 2, 0, 4, 0, 8\}$  thực hiện theo nguyên tắc trên ta được kết quả  $A[] = \{4, 4, 8, 0, 0, 0\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^5$ ;  $1 \leq A[i] \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 4 8 0 0 0
6	4 2 12 8 0 0 0 0 0 0
2 2 0 4 0 8	
10	
0 2 2 2 0 6 6 0 0 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỬA ĐÈN

### Bài làm tốt nhất

Tuyến đường ven biển của thành phố Highland có  $N$  chiếc đèn. Không may cơn bão vừa rồi đã làm hỏng  $B$  chiếc đèn.

Để khắc phục sự cố và nhanh chóng khôi phục lại hoạt động du lịch, chính quyền thành phố đã quyết định sửa tạm thời một số đèn đường bị hỏng sao cho có ít nhất một khu vực có  $K$  chiếc đèn liên tiếp hoạt động.

Các bạn hãy xác định xem số đèn đường cần phải sửa chữa ít nhất là bao nhiêu?

### Input

Dòng đầu tiên gồm 3 số nguyên dương  $N$ ,  $K$  và  $B$  ( $1 \leq B, K \leq N \leq 100\,000$ ).

$B$  dòng tiếp theo, mỗi dòng chứa vị trí của một chiếc đèn bị hỏng.

### Output

In ra số đèn đường cần sửa ít nhất sao cho có một khu vực có nhiều hơn hoặc bằng  $K$  chiếc đèn hoạt động.

### Ví dụ:

Input	Output
10 6 5 2 10 1 5 9	1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐẾM CẶP PHẦN TỬ CÓ HIỆU NHỎ HƠN K

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương và số  $k$ . Nhiệm vụ của bạn là đếm số các cặp phần tử có hiệu nhỏ hơn  $k$ . Ví dụ  $A[] = \{1, 10, 4, 2\}$ ,  $k=3$  ta nhận được kết quả là 2 tương ứng với hiệu các cặp  $(1, 2)$ ,  $(4, 2)$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$ , tương ứng với số phần tử của mảng  $A[]$  và số  $k$ ; dòng tiếp theo là  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^5$ ;  $1 \leq k \leq 10^3$ ;  $1 \leq A[i] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
4 3	3
1 10 4 2	
3 5	
2 3 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP THEO CHỮ SỐ

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên không âm. Hãy sắp đặt lại các phần tử trong mảng sao cho khi nối các số lại với nhau ta được một số lớn nhất. Ví dụ với mảng  $A[] = \{54, 546, 548, 60\}$  thực hiện theo nguyên tắc trên ta được số lớn nhất là 6054854654.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^5$ ;  $1 \leq A[i] \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6054854654
4	998764543431
54 546 548 60	
8	
1 34 3 98 9 76 45 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## BIẾN ĐỔI DÃY SỐ - 2

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên không âm. Hãy sắp đặt lại các phần tử trong mảng theo nguyên tắc:

- Phần tử đầu tiên được thay bằng tích của nó và phần tử kế tiếp.
- Phần tử cuối cùng được thay bằng tích của nó và sau nó.
- Các phần tử còn lại được thay bằng tích của phần tử sau nó và phần tử trước nó.

Ví dụ với mảng  $A[] = \{2, 3, 4, 5, 6\}$  thực hiện theo nguyên tắc trên ta được mảng  $A[] = \{6, 8, 15, 24, 30\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^5$ ;  $1 \leq A[i] \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6 8 15 24 30
5	63 108 56 72 40 30
2 3 4 5 6	
6	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## KHOẢNG CÁCH BẰNG X

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $N$  phần tử và số  $X$ . Nhiệm vụ của bạn là tìm cặp phần tử  $A[i] - A[j] = X$ . Nếu tồn tại  $A[i] - A[j] = X$  đưa ra 1, ngược lại đưa ra -1.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là cặp số  $N, X$ ; dòng tiếp theo là  $N$  số  $A[i]$  là các phần tử của mảng  $A[]$ .
- $T, N, X, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^5$ ,  $1 \leq X, A[i] \leq 10^5$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
6 78	
5 20 3 2 5 80	-1
5 45	
90 70 20 80 50	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

## PHẦN TỬ GẦN NHẤT

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $N$  phần tử đã được sắp xếp. Hãy tìm  $K$  phần tử gần nhất của  $X$  khác  $X$ . Ví dụ với mảng  $A[] = \{1, 3, 5, 7, 9, 11\}$ ,  $X = 7$ ,  $K=2$  thì ta có 2 phần tử gần nhất của 7 là 5 và 9.  $K$  là số chẵn và cần lấy một nửa lớn, một nửa nhỏ hơn  $X$ .

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm ba dòng: dòng thứ nhất đưa vào số phần tử của mảng  $N$ ; dòng tiếp theo là  $N$  số  $A[i]$  là các phần tử của mảng  $A[]$ ; dòng tiếp theo đưa vào hai số  $K$  và  $X$ .
- $T, N, K, X, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N, K, X \leq 10^6$ ,  $1 \leq A[i] \leq 10^6$  ;

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## **NHỎ NHẤT THỨ K**

### **Bài làm tốt nhất**

Cho mảng  $A[]$  gồm  $N$  phần tử. Hãy đưa ra các phần tử nhỏ nhất thứ  $k$  của mảng.

#### **Input:**

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào  $N, k$  là số phần tử của mảng  $A[]$  và số  $k$ ; các dòng tiếp theo đưa vào  $N$  số  $A[i]$ .
- $T, N, A[i], k$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq k < N \leq 10^6$ ,  $0 \leq A[i] \leq 10^6$ .

#### **Output:**

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 6 3 10 5 15 50 40 80	15

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TAM GIÁC VUÔNG

### Bài làm tốt nhất

Theo định lý Pytago, ta đã biết một bộ 3 số  $(a, b, c)$  thỏa mãn  $a^2 + b^2 = c^2$  thì đó là ba cạnh của một tam giác vuông.

Cho dãy số  $A[]$  gồm có  $N$  phần tử. Nhiệm vụ của bạn là kiểm tra xem *có tồn tại bộ ba số thỏa mãn là ba cạnh của tam giác vuông hay không*.

#### Dữ liệu vào:

- Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 20$ ).
- Mỗi test gồm số nguyên  $N$  ( $1 \leq N \leq 5000$ ).
- Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $1 \leq A[i] \leq 10^9$ ).

#### Kết quả:

Với mỗi test, in ra trên một dòng “YES” nếu tìm được, và “NO” trong trường hợp ngược lại.

#### Ví dụ:

Input	Output
2	YES
5	NO
3 1 4 6 5	
3	
1 1 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## PHÉP TOÁN VỚI PHÂN SỐ

### Bài làm tốt nhất

Phân số là sự biểu diễn số hữu tỷ dưới dạng tỷ lệ của hai số nguyên, trong đó số ở trên được gọi là tử số, còn số ở dưới được gọi là mẫu số. Cho hai phân số A và B có tử số và mẫu số được nhập từ bàn phím.

Viết chương trình thực hiện hai nhiệm vụ sau:

- Tính  $C = (A + B)^2$  và rút gọn kết quả.
- Tính  $D = A \times B \times C$  và rút gọn kết quả.

### Input:

Dòng đầu tiên là số bộ test T ( $T \leq 100$ )

T dòng tiếp theo, mỗi dòng gồm 4 số lần lượt là tử và mẫu số của phân số A và phân số B với  $-10^2 \leq$  tử số, mẫu số  $\leq 10^2$ . Mẫu số là số khác 0.

### Output:

Kết quả của hai phép tính theo định dạng phân số.

### Ví dụ

Input:	Output:
2	25/16 75/128
1 2 3 4	484/225 3872/3375
2 3 4 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main() {
    int t;
    cin >> t;
    while (t-- > 0) {
        PhanSo A;
        PhanSo B;
        cin >> A.tu >> A.mau >> B.tu >> B.mau;
        process(A, B);
    }
}
```

## DANH SÁCH SINH VIÊN - 1

### Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float).

Đọc thông tin N sinh viên từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

### Input

Dòng đầu tiên ghi số sinh viên N ( $0 < N < 50$ ).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa về dạng dd/mm/yyyy), Điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
1  Nguyen Van An  D20CQCN01-B  2/12/2002  3.19	B20DCCN001 Nguyen Van An D20CQCN01-B 02/12/2002 3.19

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct SinhVien ds[50];
    int N;
    cin >> N;
    nhap(ds, N);
    in(ds, N);
    return 0;
}
```

## DANH SÁCH SINH VIÊN - 2

### Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float).

Đọc thông tin N sinh viên từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

### Input

Dòng đầu tiên ghi số sinh viên N ( $0 < N < 50$ ).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
1  nGuyEn vaN biNH  D20CQCN01-B  2/12/2002  3.1	B20DCCN001 Nguyen Van Binh D20CQCN01-B 02/12/2002 3.10

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct SinhVien ds[50];
    int N;
    cin >> N;
    nhap(ds, N);
    in(ds, N);
    return 0;
}
```

## DANH SÁCH SINH VIÊN - 3

### Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Ngày sinh, Lớp và Điểm GPA (dạng số thực float).

Đọc thông tin N sinh viên từ bàn phím (không có mã sinh viên) sau đó sắp xếp theo điểm GPA giảm dần và in ra lần lượt màn hình mỗi dòng 1 sinh viên.

Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

### Input

Dòng đầu tiên ghi số sinh viên N ( $0 < N < 50$ ).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Dữ liệu đảm bảo không có hai sinh viên nào có điểm GPA bằng nhau.

### Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy) đã được sắp xếp theo điểm GPA giảm dần.

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
2	B20DCCN002 Nguyen Quang Hai D20DCCN02-B 01/09/1994 3.00
ngUYen Van NaM	B20DCCN001 Nguyen Van Nam D20DCCN01-B 02/12/1994 2.17



D20DCCN01-B	
2/12/1994	
2.17	
Nguyen QuanG hAi	
D20DCCN02-B	
1/9/1994	
3.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct SinhVien ds[50];
    int N;
    cin >> N;
    nhap(ds, N);
    sapxep(ds,N);
    in(ds, N);
    return 0;
}
```

## SẮP XẾP DANH SÁCH MẶT HÀNG

### Bài làm tốt nhất

Xây dựng cấp trúc mặt hàng gồm các thông tin: Mã mặt hàng (là một số nguyên, tự động tăng, tính từ 1); Tên mặt hàng, nhóm hàng: là các xâu ký tự; Giá mua, giá bán: là các số thực (không quá 9 chữ số)

Hãy sắp xếp danh sách các mặt hàng theo lợi nhuận giảm dần.

### Input:

Dòng đầu chứa số mặt hàng. Mỗi mặt hàng viết trên 4 dòng: Dòng 1: Tên mặt hàng. Dòng 2: Nhóm hàng. Dòng 3: Giá mua. Dòng 4: Giá bán

**Output:** Ghi ra danh sách mặt hàng đã sắp xếp theo lợi nhuận giảm dần (lợi nhuận tính bằng giá bán trừ đi giá mua). Mỗi mặt hàng viết trên một dòng gồm: mã, tên, nhóm hàng và lợi nhuận. Các thông tin cách nhau đúng 1 khoảng trống. Lợi nhuận viết với 2 chữ số sau dấu phẩy.

### Ví dụ:

Input	Output
3	2 Tu lanh Side by Side Dien lanh 7699.00
May tinh SONY VAIO	1 May tinh SONY VAIO Dien tu 1299.00
Dien tu	3 Banh Chocopie Tieu dung 9.50
16400	
17699	
Tu lanh Side by Side	
Dien lanh	
18300	
25999	
Banh Chocopie	
Tieu dung	
27.5	
37	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP DANH SÁCH NHÂN VIÊN

### Bài làm tốt nhất

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nu
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã), sau đó sắp xếp theo thứ tự ngày sinh từ già nhất đến trẻ nhất và in ra màn hình danh sách nhân viên đã sắp xếp.

### Input

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

### Output

Ghi ra danh sách đầy đủ nhân viên đã sắp xếp, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trống.

### Ví dụ

Input
3
Nguyen Van A
Nam
10/22/1982
Mo Lao-Ha Dong-Ha Noi
8333012345
31/12/2013
Ly Thi B

Nu
10/15/1988
Mo Lao-Ha Dong-Ha Noi
8333012346
22/08/2011
Hoang Thi C
Nu
04/02/1981
Mo Lao-Ha Dong-Ha Noi
8333012347
22/08/2011
<b>Output</b>
00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011
00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013
00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct NhanVien ds[50];
    int N,i;
    cin >> N;
    for(i = 0; i < N; i++) nhap(ds[i]);
    sapxep(ds, N);
    inds(ds, N);
    return 0;
}
```

## **BẢNG ĐIỂM THÀNH PHẦN - 1**

### **Bài làm tốt nhất**

Cho dữ liệu bảng điểm thành phần trong đó thông tin của mỗi sinh viên gồm:

- Mã sinh viên (xâu ký tự độ dài không quá 15, không có khoảng trống)
- Tên sinh viên (xâu ký tự, độ dài không quá 50)
- Lớp (xâu ký tự độ dài không quá 15, không có khoảng trống)
- Điểm 1, Điểm 2, Điểm 3: mỗi điểm là một số thực (hệ 10)

Hãy sắp xếp lại bảng điểm thành phần theo mã sinh viên (thứ tự từ điển tăng dần).

### **Input**

Dòng đầu ghi số sinh viên (không quá 100).

Mỗi sinh viên ghi trên 6 dòng lần lượt là: Mã SV, Họ tên, Lớp, Điểm 1, Điểm 2, Điểm 3.

### **Output**

Ghi ra danh sách sinh viên đã sắp xếp theo mã sinh viên.

Mỗi sinh viên ghi trên 1 dòng gồm các thông tin: thứ tự, mã sv, họ tên, lớp, điểm 1, điểm 2, điểm 3. Các thông tin cách nhau đúng một khoảng trống.

Các giá trị điểm ghi ra với đúng 1 chữ số phần thập phân.

### **Ví dụ**

<b>Input</b>	<b>Output</b>
3	1 B20DCAT001 Le Van Nam D20CQAT02-B 6.0 6.0 4.0
B20DCCN999	2 B20DCCN111 Tran Hoa Binh D20CQCN04-B 9.0 5.0 6.0
Nguyen Van An	3 B20DCCN999 Nguyen Van An D20CQCN04-B 10.0 9.0 8.0
D20CQCN04-B	
10.0	
9.0	
8.0	
B20DCAT001	
Le Van Nam	

D20CQAT02-B	
6.0	
6.0	
4.0	
B20DCCN111	
Tran Hoa Binh	
D20CQCN04-B	
9.0	
5.0	
6.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    int n;
    cin >> n;
    struct SinhVien *ds = new SinhVien[n];
    for(int i = 0; i < n; i++) {
        nhap(ds[i]);
    }
    sap_xep(ds, n);
    in_ds(ds,n);
    return 0;
}
```

## **BẢNG ĐIỂM THÀNH PHẦN - 2**

### **Bài làm tốt nhất**

Cho dữ liệu bảng điểm thành phần trong đó thông tin của mỗi sinh viên gồm:

- Mã sinh viên (xâu ký tự độ dài không quá 15, không có khoảng trống)
- Họ tên sinh viên (xâu ký tự, độ dài không quá 50)
- Lớp (xâu ký tự độ dài không quá 15, không có khoảng trống)
- Điểm 1, Điểm 2, Điểm 3: mỗi điểm là một số thực (hệ 10)

Hãy sắp xếp lại bảng điểm thành phần theo họ tên (thứ tự từ điển tăng dần - so sánh cả xâu ký tự họ tên để sắp xếp, không cần tách tên).

### **Input**

Dòng đầu ghi số sinh viên (không quá 100).

Mỗi sinh viên ghi trên 6 dòng lần lượt là: Mã SV, Họ tên, Lớp, Điểm 1, Điểm 2, Điểm 3.

### **Output**

Ghi ra danh sách sinh viên đã sắp xếp theo họ tên.

Mỗi sinh viên ghi trên 1 dòng gồm các thông tin: thứ tự, mã sv, họ tên, lớp, điểm 1, điểm 2, điểm 3. Các thông tin cách nhau đúng một khoảng trống. Các giá trị điểm ghi ra với đúng 1 chữ số phần thập phân.

### **Ví dụ**

<b>Input</b>	<b>Output</b>
3	1 B20DCAT001 Le Van An D20CQAT02-B 6.0 6.0 4.0
B20DCCN999	2 B20DCCN111 Nguyen Van Binh D20CQCN01-B 9.0 5.0 6.0
Nguyen Van Nam	3 B20DCCN999 Nguyen Van Nam D20CQCN04-B 10.0 9.0 8.0
D20CQCN04-B	
10.0	
9.0	
8.0	
B20DCAT001	



Le Van An	
D20CQAT02-B	
6.0	
6.0	
4.0	
B20DCCN111	
Nguyen Van Binh	
D20CQCN01-B	
9.0	
5.0	
6.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TRẺ NHẤT – GIÀ NHẤT

### Bài làm tốt nhất

Cho một danh sách tên người và ngày tháng năm sinh.

Hãy tìm ra người trẻ nhất và người già nhất.

### Input

Dòng 1 ghi số N là số người (không quá 100).

N dòng tiếp theo, mỗi dòng ghi tên (xâu ký tự không có khoảng trống và không quá 15 ký tự, sau đó là dãy ký tự mô tả ngày tháng năm sinh theo chuẩn dd/mm/yyyy).

### Output

Dòng đầu ghi ra tên người trẻ nhất.

Dòng thứ 2 ghi ra tên người già nhất.

Dữ liệu đảm bảo không có 2 người nào trùng ngày sinh.

### Ví dụ

Input	Output
5	Binh
Nam 01/10/1991	Tam
An 30/12/1990	
Binh 15/08/1993	
Tam 18/09/1990	
Truong 20/09/1990	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP THỜI GIAN

### Bài làm tốt nhất

Giá trị độ đo thời gian được biểu diễn bởi ba thành phần: giờ, phút, giây.

Cho N giá trị thời gian, hãy sắp xếp danh sách theo thứ tự tăng dần.

### Input

Dòng đầu ghi số nguyên dương N (không quá 5000) là số lượng giá trị thời gian cần sắp xếp.

Mỗi giá trị thời gian biểu diễn trên một dòng bằng ba số nguyên dương, lần lượt là số giờ, số phút, số giây. Trong đó số giờ đảm bảo nhỏ hơn 100, số phút và số giây đảm bảo đúng quy tắc (tức là không quá 59).

### Output

In ra danh sách đã sắp xếp theo thứ tự tăng dần.

### Ví dụ

Input	Output
3	11 15 12
11 20 20	11 20 20
14 20 14	14 20 14
11 15 12	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **DANH SÁCH THỰC TẬP - 1**

### **Bài làm tốt nhất**

Sinh viên CNTT PTIT đến năm cuối được cử đi thực tập tại các doanh nghiệp.

Thông tin của mỗi sinh viên trong danh sách thực tập bao gồm:

- Số thứ tự: là 1 số nguyên tự động tăng
- Mã sinh viên: là một chuỗi ký tự không có khoảng trống, không quá 12 ký tự
- Họ tên: là một chuỗi ký tự họ tên đã chuẩn hóa, không quá 50 ký tự
- Lớp: là một chuỗi ký tự không có khoảng trống, không quá 10 ký tự
- Email: là một địa chỉ email, không có khoảng trống, không quá 100 ký tự
- Doanh nghiệp: tên viết tắt của doanh nghiệp, không có khoảng trống, không quá 15 ký tự.

Hãy viết chương trình đọc vào danh sách thực tập sau đó in danh sách cho từng doanh nghiệp theo yêu cầu.

### **Input**

Dòng đầu ghi số N là sinh viên

Mỗi sinh viên ghi trên 5 dòng gồm mã, họ tên, lớp, email và doanh nghiệp.

Không có số thứ tự, cần tự gán theo thứ tự tăng dần từ 1.

Sau khi hết danh sách sinh viên sẽ có một số nguyên Q (không quá 5) cho biết danh sách truy vấn.

Tiếp theo là Q dòng, mỗi dòng ghi tên một doanh nghiệp (đúng như trong danh sách, không có trường hợp nào không tồn tại trong danh sách)

### **Output**

Với mỗi doanh nghiệp, liệt kê danh sách sinh viên thực tập ở doanh nghiệp đó theo thứ tự sắp xếp họ tên (so sánh cả chuỗi họ tên theo thứ tự từ điển, không cần tách riêng phần tên).

Mỗi sinh viên trên một dòng. Mỗi thông tin trong danh sách cách nhau đúng một khoảng trống.

### **Ví dụ**

Input	Output
-------	--------

6	3 B17DCAT092 Cao Danh Huy D17CQAT04-B test3@stu.ptit.edu.vn FPT
B17DCCN016	2 B17DCCN107 Dao Thanh Dat D17CNPM5 test2@stu.ptit.edu.vn FPT
Le Khac Tuan Anh	5 B17DCCN461 Dinh Quang Nghia D17CNPM2 test5@stu.ptit.edu.vn FPT
D17HTTT2	
test1@stu.ptit.edu.vn	
VIETTEL	
B17DCCN107	
Dao Thanh Dat	
D17CNPM5	
test2@stu.ptit.edu.vn	
FPT	
B17DCAT092	
Cao Danh Huy	
D17CQAT04-B	
test3@stu.ptit.edu.vn	
FPT	
B17DCCN388	
Cao Sy Hai Long	
D17CNPM2	
test4@stu.ptit.edu.vn	
VNPT	
B17DCCN461	
Dinh Quang Nghia	
D17CNPM2	
test5@stu.ptit.edu.vn	

FPT	
B17DCCN554	
Bui Xuan Thai	
D17CNPM1	
test6@stu.ptit.edu.vn	
GAMELOFT	
1	
FPT	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **DANH SÁCH THỰC TẬP - 2**

### **Bài làm tốt nhất**

Sinh viên CNTT PTIT đến năm cuối được cử đi thực tập tại các doanh nghiệp.

Thông tin của mỗi sinh viên trong danh sách thực tập bao gồm:

- Số thứ tự: là 1 số nguyên tự động tăng
- Mã sinh viên: là một chuỗi ký tự không có khoảng trống, không quá 12 ký tự
- Họ tên: là một chuỗi ký tự họ tên đã chuẩn hóa, không quá 50 ký tự
- Lớp: là một chuỗi ký tự không có khoảng trống, không quá 10 ký tự
- Email: là một địa chỉ email, không có khoảng trống, không quá 100 ký tự
- Doanh nghiệp: tên viết tắt của doanh nghiệp, không có khoảng trống, không quá 15 ký tự.

Hãy viết chương trình đọc vào danh sách thực tập sau đó in danh sách cho từng doanh nghiệp theo yêu cầu.

### **Input**

Dòng đầu ghi số N là sinh viên

Mỗi sinh viên ghi trên 5 dòng gồm mã, họ tên, lớp, email và doanh nghiệp.

Không có số thứ tự, cần tự gán theo thứ tự tăng dần từ 1.

Sau khi hết danh sách sinh viên sẽ có một số nguyên Q (không quá 5) cho biết danh sách truy vấn.

Tiếp theo là Q dòng, mỗi dòng ghi tên một doanh nghiệp (đúng như trong danh sách, không có trường hợp nào không tồn tại trong danh sách)

### **Output**

Với mỗi doanh nghiệp, liệt kê danh sách sinh viên thực tập ở doanh nghiệp đó theo thứ tự sắp xếp mã sinh viên (so sánh theo thứ tự từ điển).

Mỗi sinh viên trên một dòng. Mỗi thông tin trong danh sách cách nhau đúng một khoảng trống.

### **Ví dụ**

Input	Output
-------	--------

6	3 B17DCAT092 Cao Danh Huy D17CQAT04-B test3@stu.ptit.edu.vn FPT
B17DCCN016	2 B17DCCN107 Dao Thanh Dat D17CNPM5 test2@stu.ptit.edu.vn FPT
Le Khac Tuan Anh	5 B17DCCN461 Dinh Quang Nghia D17CNPM2 test5@stu.ptit.edu.vn FPT
D17HTTT2	
test1@stu.ptit.edu.vn	
VIETTEL	
B17DCCN107	
Dao Thanh Dat	
D17CNPM5	
test2@stu.ptit.edu.vn	
FPT	
B17DCAT092	
Cao Danh Huy	
D17CQAT04-B	
test3@stu.ptit.edu.vn	
FPT	
B17DCCN388	
Cao Sy Hai Long	
D17CNPM2	
test4@stu.ptit.edu.vn	
VNPT	
B17DCCN461	
Dinh Quang Nghia	
D17CNPM2	
test5@stu.ptit.edu.vn	



FPT	
B17DCCN554	
Bui Xuan Thai	
D17CNPM1	
test6@stu.ptit.edu.vn	
GAMELOFT	
1	
FPT	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ THUẬN NGHỊCH GIẢM DẦN

### Bài làm tốt nhất

Cho dữ liệu vào dạng văn bản, với không quá 1000 số nguyên dương, các số không quá 50 chữ số. Hãy liệt kê các số thuận nghịch khác nhau theo thứ tự giảm dần và số lần xuất hiện của nó.

### Input

Luồng vào dạng văn bản với không quá 1000 số nguyên dương. Không biết trước số dòng hoặc số lượng chính xác bao nhiêu số nguyên.

### Output

Ghi ra các số thuận nghịch theo thứ tự giá trị giảm dần và số lần xuất hiện của nó, mỗi số trên một dòng. Không tính các số có 1 chữ số.

### Ví dụ

Input	Output
12321 456 12321 34 56 99999999999999999999	99999999999999999999 1
12 3 43 34 54 34 54 34 54 657	12321 4
5 45 554 12321 12321	565 3
65 76 45 45 34 53	
546 565 65645 6 65 65 35 65	
565 565	

Giới hạn thời gian: 2s

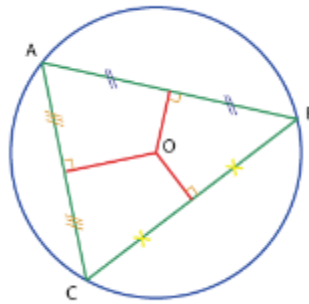
Giới hạn bộ nhớ: 65536 Kb

## DIỆN TÍCH HÌNH TRÒN NGOẠI TIẾP TAM GIÁC

### Bài làm tốt nhất

Cho ba điểm A, B, C trong không gian hai chiều 0xy.

Hãy tính diện tích hình tròn ngoại tiếp tam giác tạo bởi 3 điểm trên.



Công thức Heron tính diện tích tam giác với 3 cạnh là a, b, c:

Công thức tính bán kính hình tròn ngoại tiếp:

Khi tính diện tích nên dùng hằng số PI

```
#define PI 3.141592653589793238
```

### Input

Dòng đầu ghi số bộ test (không quá 20).

Mỗi bộ test ghi trên 1 dòng 6 số thực lần lượt là tọa độ của 3 điểm A, B, C. Giá trị tọa độ không quá 1000.

### Output

Nếu 3 điểm không thể tạo thành tam giác, in ra INVALID

Nếu 3 điểm tạo thành tam giác, in ra diện tích hình tròn ngoại tiếp với độ chính xác 3 số phân thập phân.

### Ví dụ

Input	Output
3	INVALID
0 0 0 5 0 199	INVALID
1 1 1 1 1 1	39.270
0 0 0 5 5 0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## DANH SÁCH ĐỐI TƯỢNG SINH VIÊN - 1

### Bài làm tốt nhất

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chồng toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin N thí sinh từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

### Input

Dòng đầu tiên ghi số sinh viên N ( $0 < N < 50$ ).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa về dạng dd/mm/yyyy), Điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
1  Nguyen Van An	B20DCCN001 Nguyen Van An D20CQCN01-B 02/12/2002 3.19

D20CQCN01-B	
2/12/2002	
3.19	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    SinhVien ds[50];
    int N, i;
    cin >> N;
    for(i=0;i<N;i++){
        cin >> ds[i];
    }
    for(i=0;i<N;i++){
        cout << ds[i];
    }
    return 0;
}
```

## DANH SÁCH ĐỐI TƯỢNG SINH VIÊN - 2

### Bài làm tốt nhất

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chồng toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin N thí sinh từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

### Input

Dòng đầu tiên ghi số sinh viên N ( $0 < N < 50$ ).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

### Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
1  nGuyEn    vaN        biNH	B20DCCN001 Nguyen Van Binh D20CQCN01-B 02/12/2002 3.10

D20CQCN01-B	
2/12/2002	
3.1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    SinhVien ds[50];
    int N, i;
    cin >> N;
    for(i=0;i<N;i++){
        cin >> ds[i];
    }
    for(i=0;i<N;i++){
        cout << ds[i];
    }
    return 0;
}
```



## DANH SÁCH ĐỐI TƯỢNG SINH VIÊN - 3

### Bài làm tốt nhất

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chồng toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin N thí sinh từ bàn phím (không có mã sinh viên) sau đó sắp xếp theo điểm GPA giảm dần và in ra lần lượt màn hình mỗi dòng 1 sinh viên.

Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

### Input

Dòng đầu tiên ghi số sinh viên N ( $0 < N < 50$ ).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Dữ liệu đảm bảo không có hai sinh viên nào có điểm GPA bằng nhau.

### Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy) đã được sắp xếp theo điểm GPA giảm dần.

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

### Ví dụ

Input	Output
-------	--------

2	B20DCCN002 Nguyen Quang Hai D20DCCN02-B 01/09/2002 3.00
ngUYen Van NaM	B20DCCN001 Nguyen Van Nam D20DCCN01-B 02/12/2002 2.17
D20DCCN01-B	
2/12/2002	
2.17	
Nguyen QuanG hAi	
D20DCCN02-B	
1/9/2002	
3.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    SinhVien ds[50];
    int N, i;
    cin >> N;
    for(i=0;i<N;i++){
        cin >> ds[i];
    }
    sapxep(ds, N);
    for(i=0;i<N;i++){
        cout << ds[i];
    }
    return 0;
}
```

## **DANH SÁCH ĐỐI TƯỢNG NHÂN VIÊN**

### **Bài làm tốt nhất**

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nu
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã) trong đó có sử dụng chồng toán tử nhập/xuất và in ra màn hình danh sách vừa nhập.

### **Input**

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

### **Output**

Ghi ra danh sách đầy đủ nhân viên, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trống.

### **Ví dụ**

Input
3
Nguyen Van A
Nam
10/22/1982
Mo Lao-Ha Dong-Ha Noi
8333012345
31/12/2013
Ly Thi B

Nu
10/15/1988
Mo Lao-Ha Dong-Ha Noi
8333012346
22/08/2011
Hoang Thi C
Nu
04/02/1981
Mo Lao-Ha Dong-Ha Noi
8333012347
22/08/2011
<b>Output</b>
00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013
00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011
00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    NhanVien ds[50];
    int N,i;
    cin >> N;
    for(i=0;i<N;i++) cin >> ds[i];
    for(i=0;i<N;i++) cout << ds[i];
    return 0;
}
```

## SẮP XẾP DANH SÁCH ĐỐI TƯỢNG NHÂN VIÊN

### Bài làm tốt nhất

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nu
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã) trong đó có sử dụng chồng toán tử nhập/xuất, sau đó sắp xếp theo thứ tự ngày sinh từ già nhất đến trẻ nhất và in ra màn hình danh sách đối tượng nhân viên đã sắp xếp.

### Input

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

### Output

Ghi ra danh sách đầy đủ nhân viên đã sắp xếp, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trống.

### Ví dụ

#### Input

```
3
Nguyen Van A
Nam
10/22/1982
Mo Lao-Ha Dong-Ha Noi
8333012345
31/12/2013
Ly Thi B
```

Nu
10/15/1988
Mo Lao-Ha Dong-Ha Noi
8333012346
22/08/2011
Hoang Thi C
Nu
04/02/1981
Mo Lao-Ha Dong-Ha Noi
8333012347
22/08/2011
<b>Output</b>
00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011
00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013
00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    NhanVien ds[50];
    int N,i;
    cin >> N;
    for(i=0;i<N;i++) cin >> ds[i];
    sapxep(ds, N);
    for(i=0;i<N;i++) cout << ds[i];
    return 0;
}
```

## SẮP XẾP SINH VIÊN THEO LỚP

### Bài làm tốt nhất

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trống (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trống (không quá 15)
- Email: dãy ký tự không có khoảng trống (không quá 15)

Hãy nhập danh sách sinh viên và sắp xếp theo lớp tăng dần (thứ tự từ điển)

### Input

Dòng đầu ghi số sinh viên.

Mỗi sinh viên ghi trên 4 dòng lần lượt là: mã, họ tên, lớp, email.

Có không quá 1000 sinh viên trong danh sách.

### Output

Ghi ra danh sách sinh viên đã sắp xếp theo lớp. Mỗi sinh viên trên một dòng, các thông tin cách nhau một khoảng trống.

Nếu 2 sinh viên có cùng lớp thì sắp xếp theo mã tăng dần (thứ tự từ điển)

### Ví dụ

Input	Output
4	B15DCCN215 To Ngoc Hieu D15CNPM3 sv2@stu.ptit.edu.vn
B16DCCN011	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT199 Nguyen Trong Tung D15CQKT03-B sv4@stu.ptit.edu.vn
D16CNPM1	B16DCCN011 Nguyen Trong Duc Anh D16CNPM1 sv1@stu.ptit.edu.vn
sv1@stu.ptit.edu.vn	
B15DCCN215	
To Ngoc Hieu	

D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT03-B	
sv4@stu.ptit.edu.vn	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## SẮP XẾP THEO MÃ SINH VIÊN

### Bài làm tốt nhất

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trống (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trống (không quá 15)
- Email: dãy ký tự không có khoảng trống (không quá 15)

Hãy nhập danh sách sinh viên và sắp xếp theo mã sinh viên tăng dần (thứ tự từ điển)

### Input

Mỗi sinh viên ghi trên 4 dòng lần lượt là: mã, họ tên, lớp, email.

Không cho biết số sinh viên nhưng dữ liệu đảm bảo là chẵn lần 4 dòng.

Có không quá 1000 sinh viên trong danh sách.

### Output

Ghi ra danh sách sinh viên đã sắp xếp theo mã. Mỗi sinh viên trên một dòng, các thông tin cách nhau một khoảng trống.

### Ví dụ

Input	Output
B16DCCN011	B15DCCN215 To Ngoc Hieu D15CNPM3 sv2@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
D16CNPM1	B15DCKT199 Nguyen Trong Tung D15CQKT03-B sv4@stu.ptit.edu.vn
sv1@stu.ptit.edu.vn	B16DCCN011 Nguyen Trong Duc Anh D16CNPM1 sv1@stu.ptit.edu.vn
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	

B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT03-B	
sv4@stu.ptit.edu.vn	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **LIỆT KÊ SINH VIÊN THEO LỚP**

### **Bài làm tốt nhất**

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trống (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trống (không quá 15)
- Email: dãy ký tự không có khoảng trống (không quá 15)

Hãy nhập danh sách sinh viên và liệt kê sinh viên theo lớp

### **Input**

Dòng đầu ghi số sinh viên (không quá 1000)

Mỗi sinh viên ghi trên 4 dòng lần lượt là: mã, họ tên, lớp, email.

Sau đó sẽ có giá trị số Q là số truy vấn

Tiếp theo là Q dòng, mỗi dòng ghi một lớp

### **Output**

Với mỗi truy vấn, liệt kê danh sách sinh viên của lớp đó theo mẫu như trong ví dụ. Mỗi sinh viên ghi trên một dòng, các thông tin cách nhau một khoảng trống. Thứ tự sinh viên vẫn giữ nguyên như thứ tự ban đầu.

### **Ví dụ**

Input	Output
-------	--------

4	DANH SACH SINH VIEN LOP D15CQKT02-B:
B16DCCN011	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT199 Nguyen Trong Tung D15CQKT02-B sv4@stu.ptit.edu.vn
D16CNPM1	
sv1@stu.ptit.edu.vn	
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT02-B	
sv4@stu.ptit.edu.vn	
1	
D15CQKT02-B	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ SINH VIÊN THEO KHÓA

### Bài làm tốt nhất

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trống (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trống (không quá 15)
- Email: dãy ký tự không có khoảng trống (không quá 15)

Hãy nhập danh sách sinh viên và liệt kê sinh viên theo khóa học. Chú ý: dữ liệu khóa học thể hiện qua hai chữ số thứ 2 và thứ 3 trong mã sinh viên.

### Input

Dòng đầu ghi số sinh viên (không quá 1000)

Mỗi sinh viên ghi trên 4 dòng lần lượt là: mã, họ tên, lớp, email.

Sau đó sẽ có giá trị số Q là số truy vấn

Tiếp theo là Q dòng, mỗi dòng ghi năm bắt đầu khóa học theo định dạng yyyy

### Output

Với mỗi truy vấn, liệt kê danh sách sinh viên của khóa đó theo mẫu như trong ví dụ. Mỗi sinh viên ghi trên một dòng, các thông tin cách nhau một khoảng trống. Thứ tự sinh viên vẫn giữ nguyên như thứ tự ban đầu.

### Ví dụ

Input	Output
-------	--------

4	DANH SACH SINH VIEN KHOA 2015:
B16DCCN011	B15DCCN215 To Ngoc Hieu D15CNPM3 sv2@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
D16CNPM1	B15DCKT199 Nguyen Trong Tung D15CQKT02-B sv4@stu.ptit.edu.vn
sv1@stu.ptit.edu.vn	
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT02-B	
sv4@stu.ptit.edu.vn	
1	
2015	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ SINH VIÊN THEO NGÀNH

### Bài làm tốt nhất

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trống (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trống (không quá 15)
- Email: dãy ký tự không có khoảng trống (không quá 50)

Hãy nhập danh sách sinh viên và liệt kê sinh viên theo ngành học. Chú ý: dữ liệu ngành học thể hiện qua 4 chữ cái từ thứ 4 đến thứ 7 trong mã sinh viên.

### Input

Dòng đầu ghi số sinh viên (không quá 1000)

Mỗi sinh viên ghi trên 4 dòng lần lượt là: mã, họ tên, lớp, email.

Sau đó sẽ có giá trị số Q là số truy vấn

Tiếp theo là Q dòng, mỗi dòng ghi ngành đào tạo. Chỉ có các ngành đào tạo trong danh sách sau (trong Input sẽ không có dấu):

- **Kế toán** – mã sinh viên có cụm ký tự DCKT
- **Công nghệ thông tin** – mã sinh viên có cụm DCCN – trừ đi các sinh viên lớp bắt đầu bằng chữ E
- **An toàn thông tin** – mã sinh viên có cụm DCAT – trừ các sinh viên lớp bắt đầu bằng chữ E
- **Viễn thông** – mã sinh viên có cụm DCVT
- **Điện tử** – mã sinh viên có cụm DCDT

### Output

Với mỗi truy vấn, liệt kê danh sách sinh viên của khóa đó theo mẫu như trong ví dụ. Mỗi sinh viên ghi trên một dòng, các thông tin cách nhau một khoảng trống. Thứ tự sinh viên vẫn giữ nguyên như thứ tự ban đầu.

### Ví dụ

Input	Output
-------	--------

4	DANH SACH SINH VIEN NGANH KE TOAN:
B16DCCN011	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT199 Nguyen Trong Tung D15CQKT02-B sv4@stu.ptit.edu.vn
D16CNPM1	
sv1@stu.ptit.edu.vn	
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT02-B	
sv4@stu.ptit.edu.vn	
1	
Ke toan	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## SẮP XẾP DANH SÁCH GIẢNG VIÊN

### Bài làm tốt nhất

Danh sách giảng viên Khoa CNTT cần được sắp xếp lại theo tên. Thông tin về giảng viên ban đầu chỉ có họ tên và Bộ môn. Mã giảng viên tự động tăng, tính từ GV01.

Cần sắp xếp lại theo tên (tức là từ cuối cùng trong xâu họ tên). Các giảng viên có cùng tên thì được sắp xếp theo mã giảng viên.

### Input

Dòng đầu ghi số giảng viên.

Mỗi giảng viên ghi trên 2 dòng gồm họ tên (không quá 50 ký tự) và Bộ môn (không quá 30 ký tự).

### Output

Danh sách đã sắp xếp trong đó mỗi giảng viên ghi trên một dòng. Mã được tự động điền theo thứ tự nhập, bộ môn được viết tắt theo các chữ cái đầu của từng từ và ở dạng in hoa.

### Ví dụ

Input	Ouput
3	GV02 Vu Hoai Nam KHMT
Nguyen Manh Son	GV01 Nguyen Manh Son CNPM
Cong nghe phan mem	GV03 Dang Minh Tuan ATTT
Vu Hoai Nam	
Khoa hoc may tinh	
Dang Minh Tuan	
An toan thong tin	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## DANH SÁCH GIẢNG VIÊN THEO BỘ MÔN

### Bài làm tốt nhất

Thông tin về giảng viên Khoa CNTT ban đầu chỉ có họ tên và Bộ môn. Mã giảng viên sẽ tự động điền tăng dần, tính từ GV01.

Hãy liệt kê danh sách giảng viên của Bộ môn được yêu cầu.

### **Input**

Dòng đầu ghi số giảng viên.

Mỗi giảng viên ghi trên 2 dòng gồm họ tên (không quá 50 ký tự) và Bộ môn (không quá 30 ký tự).

Tiếp theo là một dòng ghi số Q là số truy vấn.

Mỗi truy vấn là tên một bộ môn trên một dòng.

### **Output**

Danh sách các giảng viên của bộ môn theo từng truy vấn, trong đó mỗi giảng viên ghi trên một dòng. Mã được tự động điền theo thứ tự nhập, bộ môn được viết tắt theo các chữ cái đầu của từng từ và ở dạng in hoa.

Thứ tự giảng viên của mỗi bộ môn được liệt kê theo đúng thứ tự ban đầu.

### **Ví dụ**

<b>Input</b>	<b>Output</b>
3	DANH SACH GIANG VIEN BO MON CNPM:
Nguyen Manh Son	GV01 Nguyen Manh Son CNPM
Cong nghe phan mem	
Vu Hoai Nam	
Khoa hoc may tinh	
Dang Minh Tuan	
An toan thong tin	

1

Cong nghệ phần mềm

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÌM KIẾM GIẢNG VIÊN

### Bài làm tốt nhất

Thông tin về giảng viên Khoa CNTT ban đầu chỉ có họ tên và Bộ môn. Mã giảng viên sẽ tự động điền tăng dần, tính từ GV01.

Hãy tìm kiếm giảng viên theo tên (*yêu cầu tìm gần đúng – tức là trong tên giảng viên xuất hiện từ hoặc cụm từ khóa, không phân biệt chữ hoa chữ thường*).

### Input

Dòng đầu ghi số giảng viên.

Mỗi giảng viên ghi trên 2 dòng gồm họ tên (không quá 50 ký tự) và Bộ môn (không quá 30 ký tự).

Tiếp theo là một dòng ghi số Q là số truy vấn.

Mỗi truy vấn là một từ khóa cần tìm.

### Output

Danh sách các giảng viên tìm được theo từ khóa, trong đó mỗi giảng viên ghi trên một dòng. Mã được tự động điền theo thứ tự nhập, bộ môn được viết tắt theo các chữ cái đầu của từng từ và ở dạng in hoa.

Thứ tự giảng viên của mỗi bộ môn được liệt kê theo đúng thứ tự ban đầu.

### Ví dụ

Input	Output
3	DANH SACH GIANG VIEN THEO TU KHOA aN:
Nguyen Manh Son	GV01 Nguyen Manh Son CNPM
Cong nghe phan mem	GV03 Dang Minh Tuan ATTT
Vu Hoai Nam	
Khoa hoc may tinh	
Dang Minh Tuan	

An toan thong tin	
1	
aN	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## DANH SÁCH DOANH NGHIỆP NHẬN SINH VIÊN THỰC TẬP - 1

### Bài làm tốt nhất

Đề chuẩn bị cho đợt thực tập tốt nghiệp của sinh viên năm cuối, Khoa CNTT1 trao đổi với các doanh nghiệp đối tác và chốt số lượng sinh viên có thể nhận thực tập.

Hãy sắp xếp các doanh nghiệp theo số lượng sinh viên có thể nhận giảm dần.

### Input

Dòng đầu ghi số doanh nghiệp.

Mỗi doanh nghiệp ghi trên 3 dòng:

- Mã doanh nghiệp (xâu ký tự không có dấu cách, độ dài không quá 10)
- Tên doanh nghiệp (xâu ký tự độ dài không quá 150)
- Số sinh viên có thể nhận: giá trị nguyên không quá 1000

### Output

Ghi ra danh sách đã được sắp xếp theo số lượng giảm dần, mỗi thông tin ghi trên một dòng. Trong trường hợp cùng số lượng thì sắp xếp theo mã doanh nghiệp (thứ tự từ điển tăng dần).

### Ví dụ

Input
4
VIETTEL
TAP DOAN VIEN THONG QUAN DOI VIETTEL
40
FSOFT
CONG TY TNHH PHAN MEM FPT - FPT SOFTWARE
300
VNPT
TAP DOAN BUU CHINH VIEN THONG VIET NAM

200
SUN
SUN*
50
<b>Output</b>
FSOFT CONG TY TNHH PHAN MEM FPT - FPT SOFTWARE 300
VNPT TAP DOAN BUU CHINH VIEN THONG VIET NAM 200
SUN SUN* 50
VIETTEL TAP DOAN VIEN THONG QUAN DOI VIETTEL 40

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## DANH SÁCH DOANH NGHIỆP NHẬN SINH VIÊN THỰC TẬP - 2

### Bài làm tốt nhất

Đề chuẩn bị cho đợt thực tập tốt nghiệp của sinh viên năm cuối, Khoa CNTT1 trao đổi với các doanh nghiệp đối tác và chốt số lượng sinh viên có thể nhận thực tập.

Hãy lọc ra các doanh nghiệp nhận số lượng sinh viên trong đoạn  $[a,b]$ .

### Input

Dòng đầu ghi số doanh nghiệp.

Mỗi doanh nghiệp ghi trên 3 dòng:

- Mã doanh nghiệp (xâu ký tự không có dấu cách, độ dài không quá 10)
- Tên doanh nghiệp (xâu ký tự độ dài không quá 150)
- Số sinh viên có thể nhận: giá trị nguyên không quá 1000

Tiếp theo là một dòng ghi số truy vấn  $Q$ . Mỗi truy vấn là 2 số nguyên  $a, b$  viết trên một dòng trong đó  $a < b$  và dữ liệu đảm bảo luôn có ít nhất 1 doanh nghiệp nhận số lượng sinh viên trong đoạn  $[a,b]$ .

### Output

Ghi ra danh sách đã lọc trong đoạn  $[a,b]$  và được sắp xếp theo số lượng giảm dần, mỗi thông tin ghi trên một dòng. Trong trường hợp cùng số lượng thì sắp xếp theo mã doanh nghiệp (thứ tự từ điển tăng dần).

### Ví dụ

Input
4
VIETTEL
TAP DOAN VIEN THONG QUAN DOI VIETTEL
40
FSOFT
CONG TY TNHH PHAN MEM FPT - FPT SOFTWARE
300



VNPT  TAP DOAN BUU CHINH VIEN THONG VIET NAM  200  SUN  SUN*  50  1  30 50
<b>Output</b>
DANH SACH DOANH NGHIEP NHAN TU 30 DEN 50 SINH VIEN:  SUN SUN* 50  VIETTEL TAP DOAN VIEN THONG QUAN DOI VIETTEL 40

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ TỔ HỢP

### Bài làm tốt nhất

Cho số tự nhiên  $N$  và số  $K$ . Hãy đưa ra các tổ hợp chập  $K$  của  $1, 2, \dots, N$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ đôi  $N, K$  được viết trên 1 dòng.
- $T, N, K$  thỏa mãn ràng buộc:  $1 \leq T, K, N \leq 20$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	123 124 125 134 135 145 234 235 245 345
5 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ HOÁN VỊ

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Hãy đưa ra các hoán vị của  $1, 2, \dots, N$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số  $N$  được viết trên 1 dòng.
- $T, N$  thỏa mãn ràng buộc :  $1 \leq T, N \leq 20$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	123 132 213 231 312 321
3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **DÃY CON TĂNG DÀI NHẤT**

### **Bài làm tốt nhất**

Cho mảng  $A[]$  gồm  $n$  số được sinh ra ngẫu nhiên. Hãy tìm độ dài dãy tăng dài nhất các phần tử của mảng. Chú ý, dãy con của mảng không nhất thiết là liên tục. Hai phần tử giống nhau của mảng ta chỉ xem là 1 trong độ dài dãy tăng. Ví dụ với mảng  $A[] = \{5, 8, 3, 7, 9, 1\}$ , ta có kết quả là 3.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $0 \leq A[i] \leq 10^3$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6
16	3
0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15	
6	
5 8 3 7 9 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH SỐ TỔ HỢP

### Bài làm tốt nhất

Cho hai số  $n$  và  $r$ , hãy tìm  $C(n, r) \% P$ . Trong đó,  $P = 10^9 + 7$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test là bộ đôi  $n, r$ . Các số được viết cách nhau một vài khoảng trống.
- $T, n, r$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq n \leq 10^3$ ;  $1 \leq r \leq 800$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
3 2	
4 2	6

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **PHÉP CHIA DƯ CỦA TÍCH HAI SỐ**

### **Bài làm tốt nhất**

Cho ba số  $a, b, c$  có thể lớn cỡ  $10^{16}$ . Nhiệm vụ của bạn là tính  $(a*b)\%c$ .

#### **Input:**

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test trên một dòng đưa vào ba số  $a, b, c$ .
- $T, a, b, c$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 0 \leq a, b, c \leq 10^{16}$ .

#### **Output:**

- Đưa ra số kết quả mỗi test theo từng dòng.

<b>Input:</b>	<b>Output:</b>
1 8 4 5	2

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TỔNG LỚN NHẤT CỦA DÃY CON KHÔNG KÈ NHAU

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Hãy tìm tổng lớn nhất của dãy con thỏa mãn rằng buộc không có hai phần tử kề nhau thuộc một dãy con. Ví dụ với mảng  $A[] = \{3, 2, 7, 10\}$  ta nhận được kết quả là  $13 = 10 + 3$ . Với mảng  $A[] = \{3, 2, 5, 10, 7\}$  ta có kết quả là  $15 = 3 + 5 + 7$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^5$ ;  $1 \leq A[i] \leq 10^5$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	110
6	20
5 5 10 100 10 5	
3	
1 20 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TỔNG LỚN NHẤT CỦA DÃY CON LIÊN TỤC

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử. Nhiệm vụ của bạn là tìm tổng lớn nhất các dãy con liên tục của mảng  $A[]$ . Ví dụ với  $A[] = \{-2, -3, 4, -1, -2, 1, 5, -3\}$  ta có câu trả lời là 7 tương ứng với tổng lớn nhất của dãy con liên tục  $\{4, -1, -2, 1, 5\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^6$ ;  $-10^6 \leq A[i] \leq 10^6$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	9
5	-1
1 2 3 -2 5	
4	
-1 -2 -3 -4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## **DÃY CON TRUNG BÌNH LỚN NHẤT**

### **Bài làm tốt nhất**

Cho mảng  $A[]$  gồm  $n$  số và số nguyên dương  $k$ . Hãy tìm dãy con liên tục độ dài  $k$  có giá trị trung bình các phần tử lớn nhất. Ví dụ với  $A[] = \{ 1, 12, -5, -6, 50, 3 \}$  và  $k = 4$  ta có câu trả lời là  $\{12, -5, -6, 50\}$  có trung bình các phần tử lớn nhất là  $(12-5-6+30)/4=7.75$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$  và số  $k$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq k \leq n \leq 10^3$ ;  $-10^3 \leq A[i] \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	15 20
5 2	34 56
10 4 5 15 20	
4 2	
-12 34 56 7	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CÔNG 1 VÀ NHÂN ĐÔI

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Mảng  $A[]$  được gọi là mảng mục tiêu. Hãy tạo một mảng bắt đầu  $S[]$  gồm  $n$  phần tử có các phần tử ban đầu được thiết lập là 0. Chỉ được phép thực hiện hai thao tác dưới đây:

- Thao tác 1 (Increment Operation): tăng giá trị của 1 phần tử bất kỳ lên 1 đơn vị.
- Thao tác 2 (Double Operation): nhân toàn bộ phần tử trong mảng với 2.

Hãy tìm số các ít nhất để dịch chuyển mảng bắt đầu  $S[]$  thành mảng mục tiêu  $A[]$ . Ví dụ với  $A[] = \{16, 16, 16\}$  ta cần thực hiện ít nhất 7 thao tác như sau:

- Sử dụng 3 thao tác Increment để biến:  $S[0]=1, S[1]=1, S[2]=1$ .
- Sử dụng 4 thao tác Double để biến:  $S[0]=16, S[1]=16, S[2]=16$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq n \leq 10^3; 1 \leq A[i] \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
3	4
16 16 16	
2	
2 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## GIẢI MÃ TĂNG GIẢM

### Bài làm tốt nhất

Cho mảng A[] chỉ bao gồm các ký tự I hoặc D. Ký tự I được hiểu là tăng (Increasing) ký tự D được hiểu là giảm (Decreasing). Sử dụng các số từ 1 đến 9, hãy đưa ra số nhỏ nhất được đoán nhận từ mảng A[]. Chú ý, các số không được phép lặp lại. Dưới đây là một số ví dụ mẫu:

- A[] = "I" : số tăng nhỏ nhất là 12.
- A[] = "D" : số giảm nhỏ nhất là 21
- A[] = "DD" : số giảm nhỏ nhất là 321
- A[] = "DDIDDIID": số thỏa mãn 321654798

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ID
- T, Length(A) thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq \text{Length}(A) \leq 9$ ; .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
4	12
I	21
D	321
DD	321654798
DDIDDIID	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍCH LỚN NHẤT CỦA DÂY CON LIÊN TỤC

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  phần tử gồm các số âm và dương. Hãy tìm giá trị lớn nhất tích các phần tử của tất cả các dãy con liên tục trong mảng  $A[]$ . Ví dụ với mảng  $A[] = \{6, -3, -10, 0, 2\}$  ta có kết quả là 180 tương ứng với tích các phần tử của dãy con  $\{6, -3, -10\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số lượng phần tử của mảng  $N$ ; phần thứ hai đưa vào  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 100$ ;  $1 \leq A[i] \leq 200$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	180
5	120
6 -3 -10 0 2	288
6	
2 3 4 5 -1 0	
10	
8 -2 -2 0 8 0 -6 -8 -6 -1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH SỐ FIBONACCI LỚN

### Bài làm tốt nhất

Dãy số Fibonacci được định nghĩa  $F_n = F_{n-1} + F_{n-2}$ ,  $n > 1$  và  $F_0 = 0$ ,  $F_1 = 1$ . Dưới đây là một số số Fibonacci : 0, 1, 1, 2, 3, 5, 8, 13, 21...

Nhiệm vụ của bạn là tìm số Fibonacci thứ  $n$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương  $n$ .
- $T, n$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq n \leq 1000$ .

Output:

- Đưa ra kết quả mỗi test theo modulo  $10^9 + 7$  theo từng dòng.

Input	Output
2	1
2	5
5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LOẠI BỎ 100

### Bài làm tốt nhất

Cho chuỗi ký tự  $S$  chỉ bao gồm các ký tự '0' và '1'. Nhiệm vụ của bạn là loại bỏ các chuỗi con "100" trong  $S$  và đưa ra độ dài lớn nhất chuỗi con bị loại bỏ. Ví dụ  $S =$  "1011110000" ta nhận được kết quả là 6 vì ta cần loại bỏ chuỗi "110000" có độ dài 6.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test là một chuỗi ký tự nhị phân  $S$  được viết trên một dòng.
- $T, S$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq \text{Length}(S) \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
010010	6
1011110000	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍNH TỔNG

### Bài làm tốt nhất

Cho file dữ liệu dạng văn bản DATA.in có thể chứa cả số và ký tự.

Hãy lọc ra các số nguyên int trong file và tính tổng các số đó.

Chú ý: file dữ liệu có rất nhiều dòng với rất nhiều số và ký tự xen kẽ nhau. Chỉ tính tổng các số thỏa mãn điều kiện là số kiểu int.

### Input

File văn bản DATA.in có không quá 1000 dòng.

### Output

Ghi ra giá trị tổng các số tính được.

### Ví dụ

DATA.in	Output
12 3 4 5 6 7	58
Aaa 1 1 Bbb XXX yyy 5 5	
999999999999999999999999999999	
9	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ KHÁC NHAU TRONG FILE

### Bài làm tốt nhất

Cho file văn bản DATA.in có không quá 100000 số nguyên dương, giá trị các số nhỏ hơn 1000.

Hãy liệt kê các số khác nhau xuất hiện trong file và số lần xuất hiện của từng số đó.

### **Input**

File DATA.in có không quá 100000 số nguyên dương.

### **Output**

Ghi ra các số khác nhau và số lần xuất hiện theo thứ tự tăng dần

### **Ví dụ**

DATA.in	Output
17 20 25 20 15 10 24 17 25 17 22 11 23 18	10 2
14 25 12 10 12 17 21 25	11 1
	12 2
	14 1
	15 1
	17 4
	18 1
	20 2
	21 1
	22 1
	23 1
	24 1
	25 4



Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **LIỆT KÊ TỪ KHÁC NHAU**

### **Bài làm tốt nhất**

Cho file văn bản VANBAN.in.

Một từ được định nghĩa là một dãy ký tự liên tiếp không có khoảng trống, dấu tab hay dấu xuống dòng. Tạm thời chưa xét đến các dấu câu trong bài toán này.

Hãy chuyển tất cả các từ về dạng chữ thường sau đó liệt kê các từ khác nhau xuất hiện trong file VANBAN.in theo thứ tự từ điển.

### **Input**

File VANBAN.in có không quá 200 dòng.

### **Output**

Ghi ra danh sách các từ khác nhau xuất hiện trong file. Mỗi từ trên một dòng theo thứ tự từ điển.

### **Ví dụ**

<b>VANBAN.in</b>	<b>Output</b>
lap trinh Huong doi tuong	doi
lap trinh Huong thanh phan	huong
	lap
	phan
	thanh
	trinh
	tuong

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## HỢP VÀ GIAO CỦA HAI FILE VĂN BẢN

### Bài làm tốt nhất

Cho hai file DATA1.in và DATA2.in.

Một từ được định nghĩa là một dãy ký tự liên tiếp không có khoảng trống, dấu tab hay dấu xuống dòng. Tạm thời chưa xét đến các dấu câu trong bài toán này.

Hãy viết chương trình liệt kê hợp và giao của hai tập từ khác nhau trong hai file.

Các từ được liệt kê theo thứ tự từ điển.

### Input

Hai file văn bản DATA1.in và DATA2.in, có không quá 200 dòng.

### Output

Hợp và giao của hai tập từ khác nhau trong hai file ban đầu. Mỗi tập trên một dòng, các từ liệt kê theo thứ tự từ điển và cách nhau đúng một khoảng trống.

### Ví dụ

DATA1.in	Output
lap trinh huong doi tuong	ban c++ co doi huong lap ngon ngu phan thanh trinh tuong  huong lap trinh
ngon ngu lap trinh C++	
DATA2.in	
lap trinh co ban	
lap trinh huong thanh phan	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 100000 Kb

## **LỚP INTSET**

### **Bài làm tốt nhất**

Trong lý thuyết tập hợp, một tập hợp chỉ được phép chứa các giá trị phân biệt và luôn luôn lưu các giá trị theo thứ tự tăng dần.

Khai báo lớp IntSet và viết các phương thức để thực hiện các thao tác trên tập hợp số nguyên. Sử dụng lớp IntSet để in ra tập hợp các số nguyên là giao của hai tập số trong 2 dãy ban đầu.

### **Input - file văn bản DATA.in**

Dòng đầu ghi 2 số  $n$  và  $m$  ( $1 < n, m < 100$ ).

Dòng thứ 2 ghi  $n$  số của  $a[]$ .

Dòng thứ 3 ghi  $m$  số của  $b[]$ .

Các số đều dương và nhỏ hơn 1000.

### **Output**

Ghi tập giao của  $A$  và  $B$  trên một dòng theo thứ tự từ nhỏ đến lớn.

### **Ví dụ**

DATA.in	Output
5 6	3 4 5
1 2 3 4 5	
3 4 5 6 7 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## THỐNG KÊ PHẦN TỬ

### Bài làm tốt nhất

Cho 1 danh sách tuyến tính **ds** theo khai báo sau chứa n số nguyên:

```
const int MAXLIST = 10000
```

```
typedef struct list
```

```
{ int n;
```

```
    int nodes[MAXLIST];    };
```

**list ds;**

Viết chương trình con Thống kê số lần xuất hiện của từng số trong danh sách ds, và in ra màn hình mỗi số trên 1 dòng theo ví dụ sau:

Dãy số ds: 5 5 3 4 3 2 5

Kết quả in trên màn hình :

5 3

3 2

4 1

2 1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ CỦA GIAI THỪA

### Bài làm tốt nhất

Cho số tự nhiên  $N$  và số nguyên tố  $P$ . Nhiệm vụ của bạn là tìm số  $x$  lớn nhất để  $N!$  chia hết cho  $p^x$ . Ví dụ với  $N=7$ ,  $p=3$  thì  $x=2$  là số lớn nhất để  $7!$  Chia hết cho  $3^2$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số  $N, p$  được viết cách nhau một vài khoảng trống.
- $T, N, p$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^5$ ;  $2 \leq p \leq 5000$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	9
62 7	73
76 2	0
3 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## LIỆT KÊ SỐ CÓ BA ƯỚC SỐ TRONG KHOẢNG

### Bài làm tốt nhất

Cho hai số  $L, R$ . Nhiệm vụ của bạn là hãy đếm tất cả các số có đúng ba ước số trong khoảng  $[L, R]$ . Ví dụ  $L=1, R=10$ , ta có kết quả là 2 vì chỉ có số 3 và 9 là có đúng 3 ước số.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số  $L, R$ .
- $T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100; 1 \leq L, R \leq 10^{12}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
1 10	78498
1 1000000000000	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ SMITH

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy kiểm tra  $N$  có phải là số Smith hay không. Một số được gọi là số Smith nếu  $N$  không phải là số nguyên tố và có tổng các chữ số của  $N$  bằng tổng các chữ số của các ước số nguyên tố của  $N$ . Ví dụ  $N = 666$  có các ước số nguyên tố là 2, 3, 3, 37 có tổng các chữ số là 18.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên  $N$ .
- $T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N \leq 100000$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	YES
4	YES
666	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## ĐẾM SỐ NGHIỆM CỦA PHƯƠNG TRÌNH ĐỒNG DƯ

### Bài làm tốt nhất

Tìm số nghiệm của phương trình đồng dư  $x^2 = 1 \pmod{p}$  trong khoảng  $[1, b]$ . Ví dụ với  $b=5$ ,  $p=7$  ta tìm được  $x = 1 \in [1, 5]$  để  $x^2 = 1 \pmod{7}$ . Với  $b = 8$ ,  $p=6$  ta tìm được  $x = \{1, 5, 7\}$  để  $x^2 = 1 \pmod{6}$ .

$$6\%1 = 38\%1 = 34\%1 = 0; 6\%2 = 38\%2 = 34\%2 = 0; 6\%4 = 38\%4 = 34\%4 = 2;$$

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test là bộ đôi  $b, p$ . Các số được viết cách nhau một vài khoảng trống.
- $T, b, p$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $0 \leq b \leq 10^9$ ;  $1 \leq p \leq 10^5$ .

Output:

- Đưa ra số các số kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5 7	3
8 6	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## KHOẢNG CÁCH XA NHẤT

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Hãy tìm hiệu lớn nhất của  $i-j$  thỏa mãn ràng buộc  $A[i] \leq A[j]$ . Ví dụ với mảng  $A[] = \{34, 8, 10, 3, 2, 80, 30, 33, 1\}$  ta nhận được kết quả là 6 vì  $A[1] < A[7]$  và  $7-1 = 6$  là lớn nhất.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^7$ ;  $1 \leq A[i] \leq 10^8$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 9 34 8 10 3 2 80 30 33 1	6

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ĐỒNG DƯ VỚI K

### Bài làm tốt nhất

Cho mảng các số nguyên dương  $A[]$  gồm  $n$  số. Hãy tìm tất cả các số nguyên dương  $K$  sao cho tất cả các phần tử của mảng  $A[]$  lấy phần dư với  $K$  đều bằng nhau. Ví dụ với mảng  $A[] = \{6, 38, 34\}$  ta tìm được các số  $K = \{1, 2, 4\}$  vì:

$$6\%1 = 38\%1 = 34\%1 = 0; 6\%2 = 38\%2 = 34\%2 = 0; 6\%4 = 38\%4 = 34\%4 = 2;$$

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa vào số  $n$ ; dòng tiếp theo đưa vào  $n$  số của mảng  $A[]$ . Các số được viết cách nhau một vài khoảng trống.
- $T, A[i], n$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 0 \leq n \leq 10^5; 1 \leq A[i] \leq 10^5$ .

Output:

- Đưa ra số các số  $K$  kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
3	
6 38 34	1
2	
3 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP ĐẶT HAI DẪY SỐ

### Bài làm tốt nhất

Cho mảng  $A1[]$  và  $A2[]$  gồm  $n, m$  phần tử theo thứ tự. Hãy sắp xếp lại các phần tử trong  $A1[]$  theo quan hệ thứ tự trong  $A2[]$ . Phần tử xuất hiện trước trong  $A2[]$  và có mặt trước trong  $A1[]$  đứng trước; các phần tử xuất hiện trong  $A1[]$  nhưng không xuất hiện trong  $A2[]$  đứng sau theo thứ tự tăng dần. Ví dụ với mảng  $A1[] = \{2, 1, 2, 5, 7, 1, 9, 3, 6, 8, 8\}$ ,  $A2[] = \{2, 1, 8, 3\}$  sau khi sắp xếp ta được  $A1[] = \{2, 2, 1, 1, 8, 8, 3, 5, 6, 7, 9\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm ba dòng: dòng thứ nhất đưa là hai số  $n, m$ ; dòng thứ hai đưa vào  $n$  số của mảng  $A1[i]$ ; dòng thứ ba đưa vào  $m$  số của mảng  $A2[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A1[i], A2[j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n, m \leq 10^6$ ;  $1 \leq A1[i], A2[i] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 11 4 2 1 2 5 7 1 9 3 6 8 8 2 1 8 3	2 2 1 1 8 8 3 5 6 7 9

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TẬP HỢP NGUYÊN TỐ CÙNG NHAU

### Bài làm tốt nhất

Cho hai số tự nhiên  $n, m$ . Nhiệm vụ của bạn là xác định xem có thể chia các số từ 1 đến  $n$  thành hai tập sao cho giá trị tuyệt đối của tổng hai tập là  $m$  và tổng các phần tử của cả hai tập là các số đồng nguyên tố (co-prime : nguyên tố cùng nhau) hay không? Ví dụ  $n=5, m=7$  ta có kết quả là Yes vì ta chia thành 2 tập  $\{1, 2, 3, 5\}$  và  $4$  có giá trị tuyệt đối của tổng hai tập là  $7$  và là các số nguyên tố cùng nhau. Với  $n=6, m=3$  ta có câu trả lời là No vì ta có thể tìm ra hai tập  $\{1, 2, 4, 5\}$  và  $\{3, 6\}$  có trị tuyệt đối của tổng là  $3$  tuy nhiên cặp  $12=1+2+4+5$  và  $9=3+6$  không là đồng nguyên tố.

Input:

- Dòng đầu tiên đưa vào  $T$  là số lượng bộ test.
- $T$  dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là bộ hai số  $n, m$  phân biệt nhau bởi một vài khoảng trống.
- Các số  $T, n, m$ , thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq n, m \leq 10^{12}$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	Yes
5 7	No
6 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍCH CHẬP

### Bài làm tốt nhất

Phép tích chập (convolution) là kỹ thuật quan trọng trong xử lý ảnh. Kết quả phép tích chập giữa ma trận  $x[]$  và ma trận kernel  $h[]$  được xác định bằng công thức:

$$y[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k h[u, v] \cdot x[i-m, j-n]$$

Trong đó ma trận kernel có kích thước bằng  $2k+1$ . Với kernel  $3 \times 3$  thì  $-1 \leq u, v \leq 1$ , do đó, giá trị các phần tử của ma trận kết quả có dạng:

$$\begin{aligned} y[i, j] = & h[-1, -1] \cdot x[i+1, j+1] + h[-1, 0] \cdot x[i+1, j] + h[-1, 1] \cdot x[i+1, j-1] \\ & + h[0, -1] \cdot x[i, j+1] + h[0, 0] \cdot x[i, j] + h[0, 1] \cdot x[i, j-1] \\ & + h[1, -1] \cdot x[i-1, j+1] + h[1, 0] \cdot x[i-1, j] + h[1, 1] \cdot x[i-1, j-1] \end{aligned}$$

Cho ma trận ảnh và ma trận kernel  $3 \times 3$ . Nhiệm vụ của bạn là hãy thực hiện phép nhân tích chập của 2 ma trận, sau đó tính **tổng tất cả các phần tử của ma trận thu được**.

2D Convolution

2	1	0	0
3	2	1	1
4	3	2	1
2	2	1	0

\*

-1	-1	-1
-1	8	-1
-1	-1	-1

=

0	-2
7	5

Giải thích test: Vị trí ô đầu tiên của ma trận kết quả:

$$\begin{aligned} & (-1) \times 2 + (-1) \times 1 + (-1) \times 0 + \\ & (-1) \times 3 + (8) \times 2 + (-1) \times 1 + \\ & (-1) \times 4 + (-1) \times 3 + (-1) \times 2 = 0 \end{aligned}$$

### Input:

- Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 20$ ).
- Mỗi test bắt đầu bởi hai số nguyên  $N$  và  $M$ . ( $3 \leq N, M \leq 300$ ).
- Kế tiếp là  $N$  dòng, mỗi dòng gồm  $M$  số nguyên mô tả ma trận ảnh.
- 3 dòng tiếp theo, mỗi dòng gồm 3 số nguyên mô tả ma trận kernel.

- Giá trị các phần tử của hai ma trận có giá trị tuyệt đối không vượt quá 100.

### Output:

Với mỗi test, hãy in ra **tổng các phần tử của ma trận mới tìm được**.

### Ví dụ:

Input	Output
2	10
4 4	45
2 1 0 0	
3 2 1 1	
4 3 2 1	
2 2 1 0	
-1 -1 -1	
-1 8 -1	
-1 -1 -1	
3 3	
1 2 3	
4 5 6	
7 8 9	
1 1 1	
1 1 1	
1 1 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## TÍNH TOÁN TRÊN DÃY SỐ

### Bài làm tốt nhất

Cho hai hàm  $h(x)$  và  $g(x)$  xác định trên tập các số tự nhiên  $A[]$  gồm  $n$  phần tử. Trong đó,  $h(x)$  là tích của các số trong mảng  $A[]$ ,  $g(x)$  là ước số chung lớn nhất của các số trong mảng  $A[]$ . Nhiệm vụ của bạn là tìm giá trị  $h(x)^{g(x)}$ . Chú ý, khi lời giải cho kết quả lớn hãy đưa ra giá trị modulo với  $10^9+7$ .

Input:

- Dòng đầu tiên đưa vào  $T$  là số lượng bộ test.
- $T$  dòng tiếp đưa các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào số  $n$  là số các phần tử của mảng  $A[]$ ; dòng tiếp theo đưa vào  $n$  số tự nhiên phân biệt nhau bởi một vài khoảng trống.
- Các số  $T, N, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 26$ ;  $1 \leq n \leq 60$ ;  $1 \leq A[i] \leq 10^4$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	64
2	
2 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## TÍCH HAI SỐ NGUYÊN LỚN

### Bài làm tốt nhất

Cho hai số rất lớn X và Y được biểu diễn như hai chuỗi ký tự. Nhiệm vụ của bạn là tìm  $X \times Y$ ?

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa chuỗi X; dòng tiếp theo đưa vào chuỗi Y.
- T, X, Y thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $0 \leq \text{length}(X), \text{length}(Y) \leq 10^3$ .

Output:

- Đưa ra số kết quả mỗi test theo từng dòng.

Input:	Output:
1	6
2 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## CHIA HẾT CHO 8

### Bài làm tốt nhất

Cho xâu ký tự S bao gồm các số từ 0 đến 9. Nhiệm vụ của bạn là đếm số các xâu con của S là số chia hết cho 8 và không chia hết cho 3.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự số S.
- T, S thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $0 \leq \text{Length}(S) \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5
888	15
6564525600	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SẮP XẾP THEO SỐ LẦN XUẤT HIỆN

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên. Nhiệm vụ của bạn là sắp xếp mảng theo số lần xuất hiện các phần tử của mảng. Số xuất hiện nhiều lần nhất đứng trước. Nếu hai phần tử có số lần xuất hiện như nhau, số nhỏ hơn đứng trước. Ví dụ  $A[] = \{5, 5, 4, 6, 4\}$ , ta nhận được kết quả là  $A[] = \{4, 4, 5, 5, 6\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$ , tương ứng với số phần tử của mảng  $A[]$  và số  $k$ ; dòng tiếp theo là  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^4$ ;  $1 \leq k \leq 10^3$ ;  $1 \leq A[i] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 4 5 5 6
5	9 9 9 2 5
5 5 4 6 4	
5	
9 9 9 2 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ NHỎ NHẤT LỚN HƠN A[i]

### Bài làm tốt nhất

Cho mảng A[] gồm n phần tử. Nhiệm vụ của bạn là tìm giá trị nhỏ nhất lớn hơn A[i] (i=0, 1, 2,..., n-1). Đưa ra '\_' nếu A[i] không có phần tử nhỏ hơn nó. Ví dụ với mảng A[] = {13, 6, 7, 12}, ta có kết quả là {\_, 7, 12, 13}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^6$ ;  $10^6 \leq A[i] \leq 10^6$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7 6 10 9 15 3 2 _ 8
9	_ 7 12 13
6 3 9 8 10 2 1 15 7	
4	
13 6 7 12	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## PHẦN TỬ CHUNG CỦA BA DÃY SỐ

### Bài làm tốt nhất

Cho mảng  $A[]$ ,  $B[]$ ,  $C[]$  gồm  $N_1$ ,  $N_2$ ,  $N_3$  phần tử đã được sắp xếp. Hãy đưa ra các phần tử có mặt trong cả ba mảng theo thứ tự tăng dần. Nếu không có đáp án, in ra -1.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm bốn dòng: dòng thứ nhất đưa vào  $N_1$ ,  $N_2$ ,  $N_3$  là số phần tử của mảng  $A[]$ ,  $B[]$ ,  $C[]$ ; các dòng tiếp theo đưa vào  $N_1$  số  $A[i]$ ,  $N_2$  số  $B[j]$ ,  $N_3$  số  $C[k]$ .
- $T$ ,  $N_1$ ,  $N_2$ ,  $N_3$ ,  $A[i]$ ,  $B[j]$ ,  $C[k]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N_1, N_2, N_3 \leq 10^6$ ,  $0 \leq A[i], B[j], C[k] \leq 10^{18}$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 6 5 8 1 5 10 20 40 80 6 7 20 80 100 3 4 15 20 30 70 80 120	20 80

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **BỐN ĐIỂM TRÊN MẶT PHẪNG**

### **Bài làm tốt nhất**

Cho 4 điểm trong không gian 3 chiều. Nhiệm vụ của bạn là kiểm tra xem chúng có cùng nằm trên một mặt phẳng hay không? Nếu có in ra “YES”, in ra “NO” trong trường hợp ngược lại.

#### **Input:**

Dòng đầu tiên là số lượng bộ test T ( $T \leq 10000$ ).

Mỗi test gồm 4 dòng, lần lượt là tọa độ nguyên  $x[i]$ ,  $y[i]$ ,  $z[i]$  của các điểm.

( $-1000 \leq x[i], y[i], z[i] \leq 1000$ ).

#### **Output:**

Với mỗi test, in ra đáp án tìm được trên một dòng.

Input:	Output
3	YES
1 2 0	YES
2 3 0	NO
4 0 0	
0 0 0	
1 1 1	
2 2 2	
3 3 3	
4 4 4	
5 6 7	
-8 -9 -10	
12 19 0	
3 1 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## QUẢN LÝ BÁN HÀNG – 1

### Bài làm tốt nhất

Khai báo lớp Khách hàng với các thuộc tính:

- Mã khách hàng: tự động tăng, tính từ KH001
- Tên khách hàng: xâu ký tự độ dài không quá 50
- Giới tính: Nam hoặc Nu
- Ngày sinh: Theo đúng chuẩn dd/mm/yyyy
- Địa chỉ: xâu ký tự độ dài không quá 100

Khai báo lớp Mặt hàng với các thuộc tính:

- Mã mặt hàng: tự động tăng, tính từ MH001
- Tên mặt hàng: xâu ký tự độ dài không quá 100
- Đơn vị tính: xâu ký tự độ dài không quá 10
- Giá mua: số nguyên dương không quá 7 chữ số
- Giá bán: số nguyên dương không quá 7 chữ số

Khai báo lớp Hóa đơn là bạn của lớp Khách hàng và lớp Mặt hàng trong đó có các thông tin:

- Mã hóa đơn
- Mã khách hàng
- Mã mặt hàng
- Số lượng (không quá 1000)

Viết chương trình nhập danh sách hóa đơn và in danh sách ra màn hình.

### Input

Dòng đầu ghi số N là số khách hàng (không quá 20).

Tiếp theo là thông tin của N khách hàng, mỗi khách hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số M là số mặt hàng (không quá 40).

Tiếp theo là thông tin của M mặt hàng, mỗi mặt hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số K là số hóa đơn (không quá 100)

Mỗi hóa đơn ghi trên **1 dòng** gồm 3 thông tin theo đúng thứ tự đã mô tả (không có mã).

## Output

Ghi ra danh sách hóa đơn theo đúng thứ tự nhập, trong đó gồm các thông tin sau, mỗi thông tin cách nhau đúng một khoảng trống.

- Mã hóa đơn
- Tên khách hàng
- Địa chỉ
- Tên mặt hàng
- Đơn vị tính
- Giá mua
- Giá bán
- Số lượng
- Thành tiền

## Ví dụ

Input
2
Nguyen Van Nam
Nam
12/12/1997
Mo Lao-Ha Dong-Ha Noi
Tran Van Binh
Nam
11/14/1995
Phung Khoang-Nam Tu Liem-Ha Noi
2
Ao phong tre em
Cai
25000

41000

Ao khoac nam

Cai

240000

515000

3

KH001 MH001 2

KH001 MH002 3

KH002 MH002 4

### Output

HD001 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao phong tre em Cai 25000 41000 2 82000

HD002 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao khoac nam Cai 240000 515000 3 1545000

HD003 Tran Van Binh Phung Khoang-Nam Tu Liem-Ha Noi Ao khoac nam Cai 240000 515000 4 2060000

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    KhachHang dskh[25];
    MatHang dsmh[45];
    HoaDon dshd[105];
    int N,M,K,i;
    cin >> N;
    for(i=0;i<N;i++) cin >> dskh[i];
    cin >> M;
    for(i=0;i<M;i++) cin >> dsmh[i];
    cin >> K;
    for(i=0;i<K;i++) cin >> dshd[i];

    for(i=0;i<K;i++) cout << dshd[i];
    return 0;
}
```

## QUẢN LÝ BÁN HÀNG – 2

### Bài làm tốt nhất

Khai báo lớp Khách hàng với các thuộc tính:

- Mã khách hàng: tự động tăng, tính từ KH001
- Tên khách hàng: xâu ký tự độ dài không quá 50
- Giới tính: Nam hoặc Nu
- Ngày sinh: Theo đúng chuẩn dd/mm/yyyy
- Địa chỉ: xâu ký tự độ dài không quá 100

Khai báo lớp Mặt hàng với các thuộc tính:

- Mã mặt hàng: tự động tăng, tính từ MH001
- Tên mặt hàng: xâu ký tự độ dài không quá 100
- Đơn vị tính: xâu ký tự độ dài không quá 10
- Giá mua: số nguyên dương không quá 7 chữ số
- Giá bán: số nguyên dương không quá 7 chữ số

Khai báo lớp Hóa đơn là bạn của lớp Khách hàng và lớp Mặt hàng trong đó có các thông tin:

- Mã hóa đơn
- Mã khách hàng
- Mã mặt hàng
- Số lượng (không quá 1000)
- Lợi nhuận

Viết chương trình nhập danh sách hóa đơn, sắp xếp theo lợi nhuận giảm dần và in danh sách ra màn hình.

### Input

Dòng đầu ghi số N là số khách hàng (không quá 20).

Tiếp theo là thông tin của N khách hàng, mỗi khách hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số M là số mặt hàng (không quá 40).

Tiếp theo là thông tin của M mặt hàng, mỗi mặt hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số K là số hóa đơn (không quá 100)

Mỗi hóa đơn ghi trên **1 dòng** gồm 3 thông tin theo đúng thứ tự đã mô tả (không có mã và lợi nhuận).

## Output

Ghi ra danh sách hóa đơn đã sắp xếp, trong đó gồm các thông tin sau, mỗi thông tin cách nhau đúng một khoảng trống.

- Mã hóa đơn
- Tên khách hàng
- Địa chỉ
- Tên mặt hàng
- Số lượng
- Thành tiền
- Lợi nhuận

## Ví dụ

Input
2
Nguyen Van Nam
Nam
12/12/1997
Mo Lao-Ha Dong-Ha Noi
Tran Van Binh
Nam
11/14/1995
Phung Khoang-Nam Tu Liem-Ha Noi
2
Ao phong tre em
Cai
25000

41000

Ao khoac nam

Cai

240000

515000

3

KH001 MH001 2

KH001 MH002 3

KH002 MH002 4

### Output

HD003 Tran Van Binh Phung Khoang-Nam Tu Liem-Ha Noi Ao khoac nam 4 2060000 1100000

HD002 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao khoac nam 3 1545000 825000

HD001 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao phong tre em 2 82000 32000

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    KhachHang dskh[25];
    MatHang dsmh[45];
    HoaDon dshd[105];
    int N,M,K,i;
    cin >> N;
    for(i=0;i<N;i++) cin >> dskh[i];
    cin >> M;
    for(i=0;i<M;i++) cin >> dsmh[i];
    cin >> K;
    for(i=0;i<K;i++) cin >> dshd[i];

    sapxep(dshd, K);

    for(i=0;i<K;i++) cout << dshd[i];
    return 0;
}
```

## ĐẾM SỐ CÁCH DI CHUYỂN

### Bài làm tốt nhất

Cho ma trận vuông  $A[N][N]$  gồm các số nguyên dương và số tự nhiên  $K$ . Hãy tìm số các cách di chuyển từ phần tử đầu tiên ( $A[0][0]$  đến phần tử cuối cùng  $A[N-1][N-1]$ ) sao cho tổng các phần tử của phép di chuyển đúng bằng  $K$ . Biết từ phần tử  $A[i][j]$ , ta chỉ được phép dịch chuyển đến phần tử  $A[i+1][j]$  hoặc  $A[i][j+1]$ . Ví dụ với ma trận dưới đây sẽ có 2 phép di chuyển theo nguyên tắc kể trên để có tổng bằng 12.

1      2      3

4      6      5      Cách 1:  $1 \rightarrow 2 \rightarrow 6 \rightarrow 2 \rightarrow 1$

3      2      1      Cách 2:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 1$

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số  $N, K$ ; dòng tiếp là  $N \times N$  các phần tử của ma trận  $A[][]$ ; các phần tử được viết cách nhau một vài khoảng trống.
- $T, N, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 20$ ;  $1 \leq A[i][j] \leq 200$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	0
3 16	2
1 2 3	
4 6 5	
9 8 7	
3 12	
1 2 3	
4 6 5	
3 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **DÃY SỐ BITONIC**

### **Bài làm tốt nhất**

Cho mảng  $A[]$  gồm  $n$  phần tử gồm các số nguyên dương. Mảng  $A[]$  được gọi là Bitonic nếu các phần tử của mảng được chia thành hai phần: phần thứ nhất tăng dần, phần thứ hai giảm dần. Mảng  $A[]$  được sắp xếp tăng dần cũng là mảng Bitonic khi xem phần thứ hai là rỗng. Tương tự như vậy, mảng  $A[]$  được sắp xếp giảm dần cũng là một bitonic. Hãy tìm độ dài dãy con dài nhất của mảng  $A[]$  là một Bitonic. Ví dụ với mảng  $A[] = \{1, 11, 2, 10, 4, 5, 2, 1\}$  ta có kết quả là 6 tương ứng với độ dài dãy con  $\{1, 2, 10, 4, 2, 1\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số lượng phần tử của mảng  $N$ ; phần thứ hai đưa vào  $n$  số  $A[i]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 100$ ;  $1 \leq A[i] \leq 200$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5
5	6
1 2 5 3 2	
8	
1 11 2 10 4 5 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## TỔNG LỚN NHẤT CỦA DÃY CON TĂNG DẦN

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên dương. Hãy tìm tổng lớn nhất của dãy con tăng dần của dãy số  $A[]$ . Ví dụ với  $A[] = \{ 1, 101, 2, 3, 100, 4, 5 \}$  ta có câu trả lời là  $106=1+2+3+100$ . Với dãy  $A[] = \{10, 5, 4, 3\}$  ta có câu trả lời là 10.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^3$ ;  $1 \leq A[i] \leq 10^7$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	106
7	10
1 101 2 3 100 4 5	
4	
10 5 4 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## ƯỚC SỐ CHUNG LỚN NHẤT CỦA SỐ NGUYÊN LỚN

### Bài làm tốt nhất

Cho hai số  $a$  và  $b$  trong đó  $a \leq 10^{12}$ ,  $b \leq 10^{250}$ . Nhiệm vụ của bạn là tìm ước số chung lớn nhất của hai số  $a$ ,  $b$ .

Input:

- Dòng đầu tiên đưa vào  $T$  là số lượng bộ test.
- $T$  dòng tiếp đưa các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào số  $a$ ; dòng tiếp theo đưa vào số  $b$ .
- Các số  $T$ ,  $a$ ,  $b$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq a \leq 10^{12}$ ;  $1 \leq b \leq 10^{250}$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	3
1221	
1234567891011121314151617181920212223242526272829	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ HOÀN HẢO

### Bài làm tốt nhất

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy kiểm tra  $N$  có phải là số hoàn hảo hay không. Một số  $N$  được gọi là số hoàn hảo nếu tổng các ước số của nó bằng chính nó. Ví dụ  $N = 6 = 1 + 2 + 3$  là số hoàn hảo.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên  $N$ .
- $T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100; 1 \leq N \leq 10^{18}$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
6	0
21	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## SỐ LỚN NHẤT CỦA DÃY CON LIÊN TỤC

### Bài làm tốt nhất

Cho mảng  $A[]$  gồm  $n$  số nguyên không âm và số  $k$ . Hãy tìm số lớn nhất của mỗi dãy con liên tục gồm  $k$  phần tử của mảng. Ví dụ với mảng  $A[] = \{1, 2, 3, 1, 4, 5, 2, 3, 6\}$ ,  $K = 3$ , ta có kết quả  $3\ 3\ 4\ 5\ 5\ 6$ .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $n$  là số phần tử của mảng  $A[]$  và số  $k$ ; dòng kế tiếp đưa vào  $n$  số  $A[i]$  của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq k < n \leq 10^7$ ;  $0 \leq A[i] \leq 10^7$ ;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3 3 4 5 5 5 6
9 3	10 10 10 15 15 90 90
1 2 3 1 4 5 2 3 6	
10 4	
8 5 10 7 9 4 15 12 90 13	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Input:	Output:
2	10 11 7 6 5 9 13 14 15 16 12 8 4 3 2 1
4 4	11 10 9 8 7 13 14 15 16 17 18 12 6 5 4 3 2
1 2 3 4	1
5 6 7 8	
9 10 11 12	
13 14 15 16	
3 6	
1 2 3 4 5 6	
7 8 9 10 11 12	

13 14 15 16 17 18

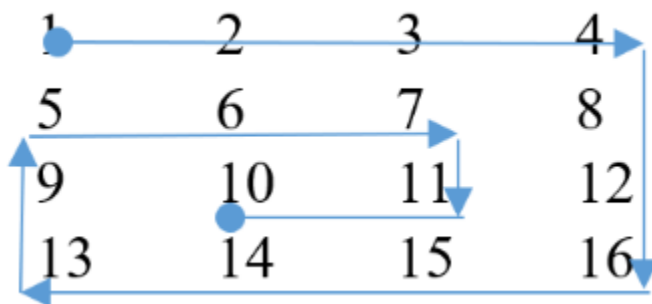
Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## MA TRẬN XOẮN ỐC - 3

### Bài làm tốt nhất

Cho ma trận  $A[N][M]$ . Nhiệm vụ của bạn là đưa ra phần tử thứ  $k$  phép duyệt theo mô hình xoắn ốc trên ma trận của ma trận theo hình xoắn ốc. Ví dụ với  $k=6$  của ma trận dưới đây sẽ cho ta kết quả là 12 ( Phép duyệt xoắn ốc: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10 ).



Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào  $N, M$  là cấp của ma trận  $A[][]$  và số  $k$ ; dòng tiếp theo đưa vào  $N \times M$  số  $A[i][j]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, M, N, k, A[i][j]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq M, N, k \leq 100$ ;  $1 \leq A[i][j] \leq 10^5$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	12
4 4 6	5
1 2 3 4	
5 6 7 8	
9 10 11 12	
13 14 15 16	
3 4 10	
1 2 3 4	
5 6 7 8	
9 10 11 12	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb



## MẬT KHẨU

### Bài làm tốt nhất

(Giới hạn thời gian chạy: 10 giây)

Hệ thống quản lý đào tạo của PTIT đang gặp một vấn đề về bảo mật. Do sự cố này, các account bị đổi thành tên viết liền của các sinh viên. Và chỉ cần đánh một chuỗi kí tự có chứa mật khẩu là có thể đăng nhập vào hệ thống. Chẳng hạn sinh viên A có mật khẩu là “abcd”, nếu ai đó đăng nhập với tài khoản là tên của A, mật khẩu “abcdef” hay “aaaabcd” đều được chấp nhận.

Nhân cơ hội này, rất nhiều bạn sinh viên đã cố gắng hack vào tài khoản của những người khác. Cho biết danh sách mật khẩu của tất cả các user, bài toán đặt ra là hãy xác định xem có nhiều nhất bao nhiêu trường hợp user này có thể login vào user khác?

### Input:

Dòng đầu tiên là số nguyên  $N$  ( $1 \leq N \leq 100\,000$ ).

$N$  dòng tiếp theo, mỗi dòng chứa mật khẩu của một user, có độ dài không quá 10 kí tự và chỉ gồm các kí tự thường.

### Output:

In ra một số nguyên là đáp án tìm được.

### Ví dụ:

Test 1	Test 2
Input :  3 aaa  aa  abb   Output :  1	Input :  3  x x xy   Output :  4

--	--

Giải thích test 2: User 1 có thể login vào user 2 và ngược lại. User 3 có thể login vào tài khoản của user 1 và 2.

Giới hạn thời gian: 10s

Giới hạn bộ nhớ: 65536 Kb

## DIỆN TÍCH ĐA GIÁC

### Bài làm tốt nhất

Cho một đa giác lồi có  $N$  đỉnh trên mặt phẳng Oxy.

Nhiệm vụ của bạn là hãy tính diện tích đa giác này.

#### Input:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 100$ ).

Mỗi test bắt đầu bởi số nguyên  $N$  ( $N \leq 1000$ ).

$N$  dòng tiếp theo, mỗi dòng gồm 2 số nguyên  $x[i]$ ,  $y[i]$  ( $-1000 \leq x[i]$ ,  $y[i] \leq 1000$ ) là tọa độ của điểm thứ  $i$ . Các điểm được liệt kê theo thứ tự ngược chiều quay kim đồng hồ.

#### Output:

Với mỗi test, in ra đáp án tìm được trên một dòng.

Input:	Output
2	0.500
3	4.000
0 0	
1 0	
0 1	
4	
0 0	
2 0	
2 2	
0 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **DÃY CON LIÊN TỤC NHỎ NHẤT**

### **Bài làm tốt nhất**

Cho mảng  $A[]$  gồm  $n$  số nguyên và số  $X$ . Hãy tìm độ dài dãy con liên tục nhỏ nhất có tổng lớn hơn  $X$ . Ví dụ với  $A[] = \{1, 4, 45, 6, 0, 19\}$  và  $X = 51$  ta có câu trả lời là 3 tương ứng với dãy con  $\{4, 45, 6\}$ . Với dãy  $A[] = \{1, 10, 5, 2, 7\}$  và  $X = 9$  ta có câu trả lời là 1 tương ứng với dãy con  $\{10\}$ . Với dãy  $A[] = \{1, 2, 4\}$  và  $X = 8$  ta có câu trả lời là -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T$ .
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng  $n$  và số  $X$ ; dòng tiếp theo là  $n$  số  $A[i]$  của mảng  $A[]$ ; các số được viết cách nhau một vài khoảng trống.
- $T, n, X, A[i]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq n \leq 10^7$ ;  $1 \leq A[i] \leq 10^7$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
6 51	-1
1 4 45 6 0 19	
3 8	
1 2 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

## **BẢNG MÀU R - G - B**

### **Bài làm tốt nhất**

Ta cần tạo một xâu ký tự S có độ dài n. Trong đó, mỗi ký tự trong S chỉ là các ký tự R, B, hoặc G. Xâu ký tự nhận được có ít nhất r ký tự R, b ký tự B, g ký tự G ( $r + b + g \leq n$ ). Hãy đếm số các xâu ký tự thỏa mãn yêu cầu kể trên. Ví dụ với  $n=4, r=1, b=1, g=1$  ta có thể có 36 xâu ký tự khác nhau.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là bộ bốn số phân biệt n, r, b, g được viết trên một dòng.
- T, S, n, r, b, g thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq n \leq 20; 1 \leq r, b, g \leq N$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	36
4 1 1 1	22
4 2 0 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

