

# Генерація та виведення масиву дійсних чисел

Написати програму для впорядкування масиву рядків, використовуючи покажчики.

Основна таблиця ASCII

	00	10	20	30	40	50	60	70
0		▸		0	@	P	'	p
1	☒	◀	!	1	A	Q	a	q
2	☒	‡	"	2	B	R	b	r
3	♥	!!	#	3	C	S	c	s
4	♦	¶	\$	4	D	T	d	t
5	♣	§	%	5	E	U	e	u
6	♠	=	&	6	F	V	f	v
7	•	±	'	7	G	W	g	w
8	▣	↑	<	8	H	X	h	x
9	○	↓	)	9	I	Y	i	y
A	☒	→	*	:	J	Z	j	z
B	♂	←	+	;	K	[	k	{
C	♀	└	,	<	L	\	l	:
D	℞	↕	-	=	M	]	m	}
E	☾	▲	.	>	N	^	n	~
F	※	▼	/	?	O	_	o	△

Розширена таблиця ASCII (ср866)

	80	90	A0	B0	C0	D0	E0	F0
0	А	Р	а	▣	└	└	р	≡
1	Б	С	б	▣	└	└	с	±
2	В	Т	в	▣	└	└	т	>
3	Г	У	г		└	└	у	<
4	Д	Ф	д	└	-	└	ф	└
5	Е	Х	е	└	└	└	х	└
6	Ж	Ц	ж	└	└	└	ц	÷
7	З	Ч	з	└	└	└	ч	≈
8	И	Ш	и	└	└	└	ш	°
9	Й	Щ	й	└	└	└	щ	-
A	К	Ь	к	└	└	└	ь	-
B	Л	Ы	л	└	└	▣	ы	√
C	М	Ь	м	└	└	▣	ь	n
D	Н	Э	н	└	=	▣	э	2
E	О	Ю	о	└	└	▣	ю	●
F	П	Я	п	└	└	▣	я	

48-57	-----	0..9
65-90	-----	A..Z
97-122	-----	a..z
128-159	-----	A..Я
160-175	-----	а..п
224-239	-----	р..я

```

void main(void)
{ char **str; // покажчик на масив рядків
  int a,b,nums,numс, ch;
  clrscr();
  /* for (a=1;a<240;a++){ if (a%10) cout<<a<<"-> "<<(char)a<<" \n";
  else getch();}*/ // виведення таблиці символів ср866
  cout<<"Введіть кількість рядків "; cin>>nums;
  cout<<"Введіть кількість символів в рядку";cin>>numс;
  str=(char**) calloc(nums, sizeof(char*));
  for(a=0;a<nums;a++) str[a]=(char*) calloc(numс+1, sizeof(char));
  randomize();
  for(a=0;a<nums;a++){ b=0;
  do{
    ch=random(257);
    if ( (ch>64 && ch<91) || (ch>96 && ch<123) ) // генерація символів рядку
    {
      str[a][b]= ch;
      b++;
    } //for
  } while (b<numс); //do
  str[a][b]= '\0'; // вставка символу кінця рядку
  }
}

```

## Комп'ютерний практикум № 7

### Завдання:

Написати програму для обчислення коренів нелінійних рівнянь на заданому проміжку

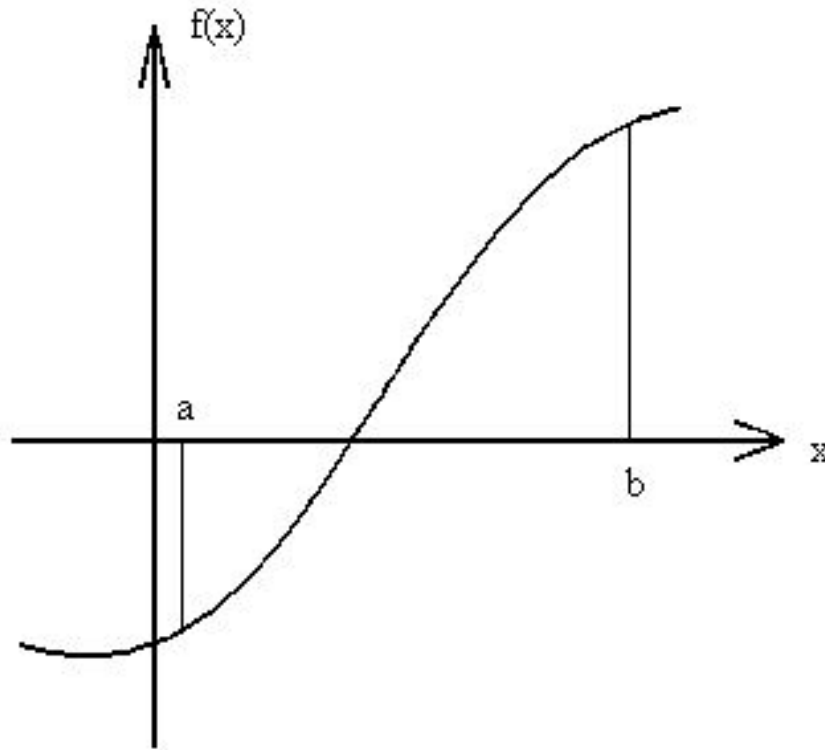
$$\cos \frac{y}{x} - 2 \sin \frac{1}{x} + \frac{1}{x} = 0, x \in [a1; a2];$$

$$\sin(\ln x) - \cos(\ln x) + y \ln x = 0, x \in [a1; a2]$$

Корені можна обчислити:

1. методом половинного ділення
2. методом дотичних (Ньютона)

# Метод половинного ділення



На кожному кроці обчислюється значення і перевіряється знак добутку

$$x = \frac{a + b}{2}$$

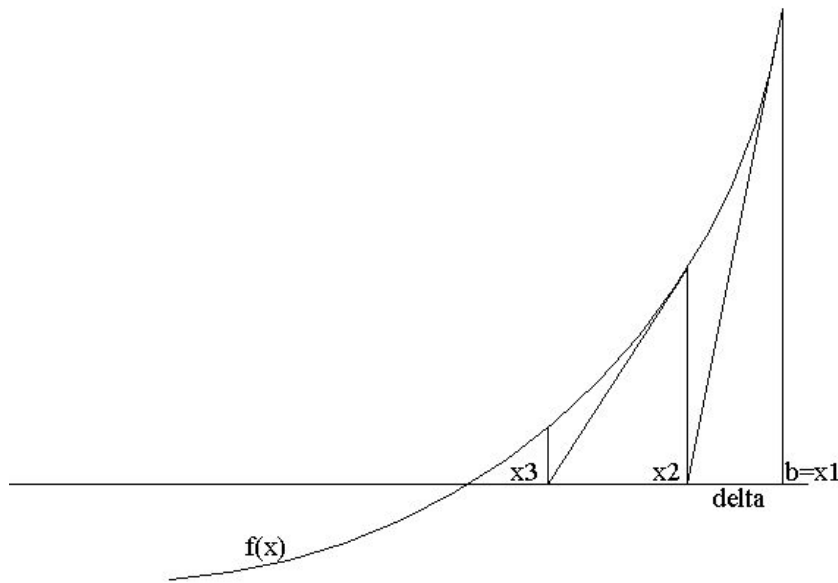
$f(a) \cdot f(x) > 0$ , то відбувається

присвоєння  $a = x$ , в протилежному випадку

$$b = x.$$

Ці дії повторюються поки  $|b - a| > \varepsilon$ , де  $\varepsilon$  – наперед задана точність.

# Метод дотичних (Ньютона)



1. Знаходиться значення функції в крайній точці –  $f(b)$ ;
2. Через знайдену точку проводимо дотичну. Знаходимо точку  $x_2$ .  
Алгебраїчно це виглядає наступним чином:

$$delta = \frac{f(x_i)}{f'(x_i)}; \quad x_{i+1} = x_i - delta;$$

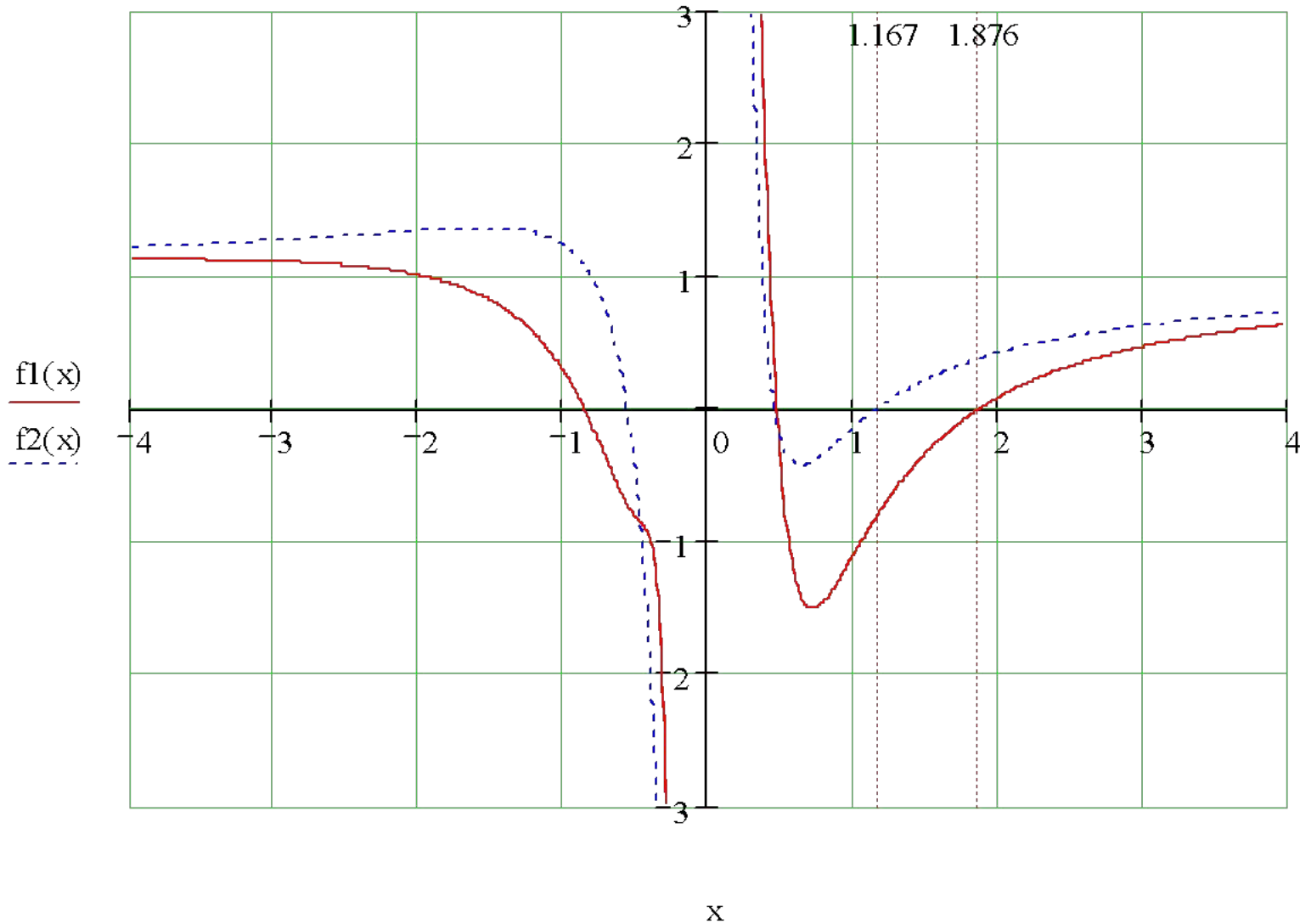
$f'(x_i)$  - похідна функції в точці  $x_i$

3. Повторюємо п.2 доти, поки:  $|delta| > eps$ .

$$f'(x_i) = \frac{f(x_i + \varepsilon) - f(x_i)}{\varepsilon}, \quad \varepsilon - \text{додатнє, близьке до нуля число (підбирається).}$$

$$f1(x) = \cos \frac{y}{x} - 2 \sin \frac{1}{x} + \frac{1}{x} = 0, x \in [a1;a2], y=2 \quad \cos \frac{2}{x} - 2 \sin \frac{1}{x} + \frac{1}{x} = 0, x \in [1;2]$$

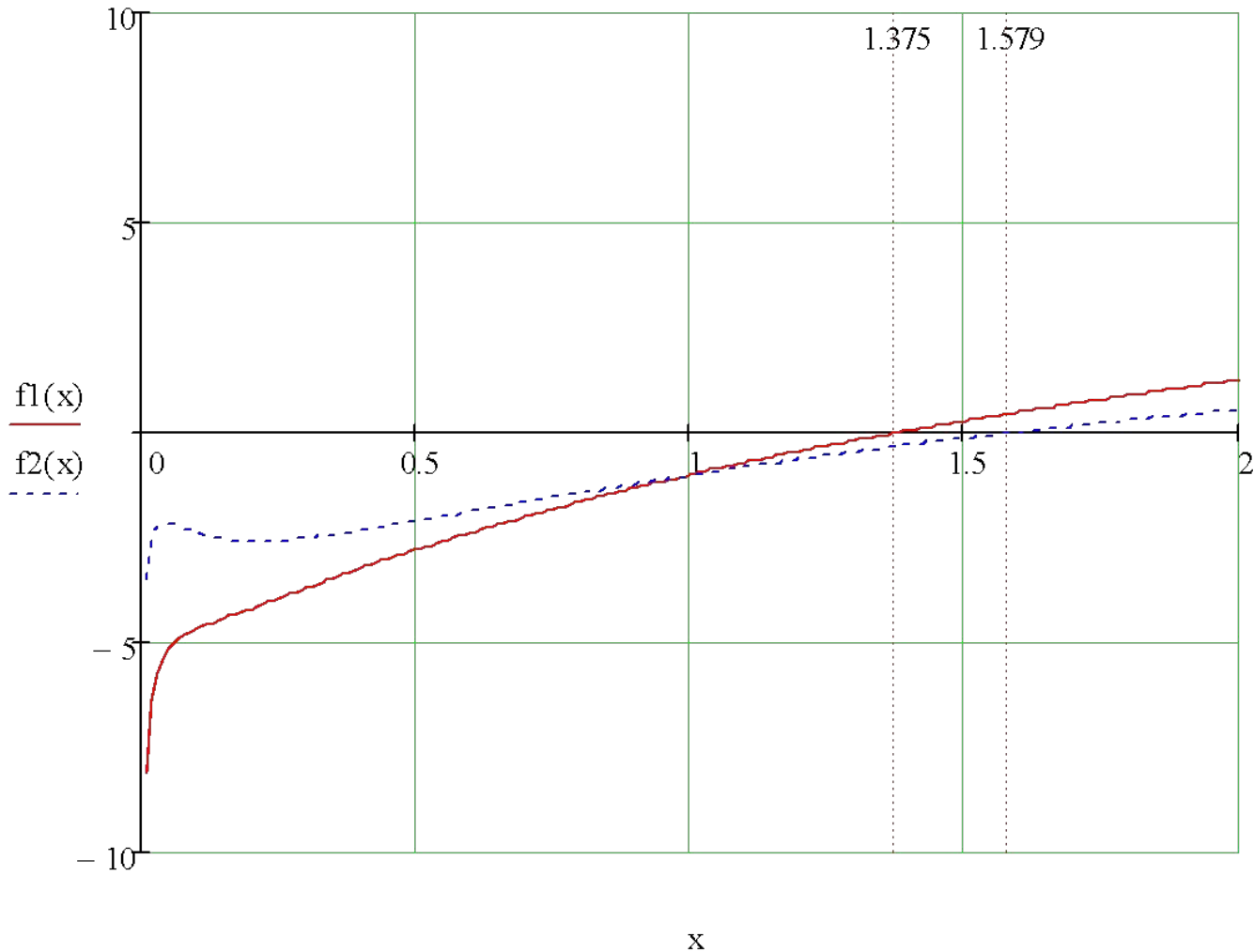
$$f2(x) = \cos \frac{y}{x} - 2 \sin \frac{1}{x} + \frac{1}{x} = 0, x \in [a1;a2], y=1 \quad \cos \frac{1}{x} - 2 \sin \frac{1}{x} + \frac{1}{x} = 0, x \in [1;2]$$



$$\sin(\ln x) - \cos(\ln x) + y \ln x = 0, x \in [a1; a2]$$

$$f1(x): x \in [1; 2], y=2 \quad \boxed{x = 1.375}$$

$$f2(x): x \in [1; 2], y=1 \quad \boxed{x = 1.579}$$





У деяких випадках треба мати аналог функції – змінної, наприклад, в залежності від певних умов обчислювати різні функції, або написати функцію інтегрування певного класу функцій.

Тобто, потрібна функціональна змінна, яка реалізується за допомогою покажчика на функцію.

Якщо є кілька функцій певної структури, наприклад:

```
double fun1 (double, int);
```

```
...
```

```
double fun3 (double, int);
```

то покажчик на такі функції матиме вигляд:

```
double (*fpr) (double, int);
```

Тут дужки (*\*fpr*) потрібні, бо модифікатор ( ) має перевагу над модифікатором \*. Тому:

```
double *fpr (double, int); означатиме функцію,
```

яка передає **покажчик** на *double*.

За допомогою покажчиків на функцію, можна створювати масиви функцій, використовувати їх як параметри інших функцій.

Наприклад, для числового інтегрування зазначених вище функцій можна створити універсальну функцію, яка має параметрами:

1. Межі інтегрування;
2. Точність;
3. Функцію, для якої обчислюється інтеграл.

```
double integrate (double (*fpr)(double), double lowbow,  
double highbow, double eps);
```



.....  
// функція знаходження корня

```
float kor(float (*fpr)(float), float x, float y, float eps);
```

// функція знаходження значення в точці x

```
float fun1 (float x) {return (cos(2.0/x)-2*sin(1.0/x)+1.0/x);}
```

// функція знаходження значення в точці x

```
float fun2 (float x) {return (sin(log(x))-cos(log(x))+2*log(x));}
```

.....

```
float kor ( float (*fpr)(float), float l, float h, float eps)
```

```
{ float x;
```

```
  while (fabs(h-l)>eps)
```

```
  { x=(l+h)/2.0;
```

```
    ( (*fpr)(l) * (*fpr)(x) > 0 ) ? (l=x) : (h=x);
```

```
  }
```

```
  return x;
```

```
}
```

