

## ZADÁNÍ SEMESTRÁLNÍ PRÁCE

### VIZUALIZACE GRAFU MATEMATICKÉ FUNKCE

---

#### Zadání

Naprogramujte v ANSI C přenositelnou<sup>1</sup> **konzolovou aplikaci**, která jako vstup načte z parametru na příkazové řádce matematickou funkci ve tvaru  $y = f(x); x, y \in \mathbb{R}$ , provede její analýzu a vytvoří soubor ve formátu PostScript s grafem této funkce na zvoleném definičním oboru.

Program se bude spouštět příkazem `graph.exe <func> <out-file> [<limits>]`. Symbol `<func>` zastupuje zápis matematické funkce jedné reálné nezávisle proměnné (funkce ve více dimenzích program řešit nebude, nalezne-li během analýzy zápisu funkce více nezávisle proměnných než jednu, vypíše srozumitelné chybové hlášení a skončí). Závisle proměnná (zde  $y$ ) je implicitní, a proto se levá strana rovnosti v zápisu nebude uvádět. Symbol `<out-file>` zastupuje jméno výstupního postscriptového souboru. Takže Váš program může být během testování spuštěn například takto:

```
X:\>graph.exe "sin(2*x)^3" mygraph.ps
```

Výsledkem práce programu bude soubor ve formátu PostScript, který bude zobrazovat graf zadané matematické funkce – ve výše uvedeném případě  $y = \sin(2x)^3$  – v kartézské souřadnicové soustavě  $(O; x, y)$  s vyznačenými souřadnými osami a (aspoň) význačnými hodnotami definičního oboru a oboru hodnot funkce (viz Specifikace výstupu programu).

Pokud nebudou na příkazové řádce uvedeny alespoň dva argumenty, vypíše chybové hlášení a stručný návod k použití programu v angličtině podle běžných zvyklostí (viz např. ukázková semestrální práce na webu předmětu Programování v jazyce C). **Vstupem programu jsou pouze argumenty na příkazové řádce – interakce s uživatelem pomocí klávesnice či myši v průběhu práce programu se neočekává.**

Hotovou práci odevzdejte v jediném archivu typu ZIP prostřednictvím automatického odevzdávacího a validačního systému. Postupujte podle instrukcí uvedených na webu předmětu. Archiv nechtě obsahuje všechny zdrojové soubory potřebné k přeložení programu, **makefile** pro Windows i Linux (pro překlad v Linuxu připravte soubor pojmenovaný **makefile** a pro Windows **makefile.win**) a dokumentaci ve formátu PDF vytvořenou v typografickém systému  $\text{T}_{\text{E}}\text{X}$ , resp.  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . **Bude-li některá z částí chybět, validační systém Vaši práci odmítne.**

Podrobné informace k odevzdání práce najdete na webové stránce předmětu Programování v jazyce C na adrese URL <https://www.kiv.zcu.cz/studies/predmety/pc/index.php#work>.

#### Specifikace vstupu programu

Vstupem programu jsou pouze parametry na příkazové řádce: Prvním (a nejdůležitějším) parametrem je matematická funkce, kterou má program zpracovat. S ohledem na to, že může její zápis obsahovat mezery, napište program tak, aby akceptoval jak zápis funkce obklopený uvozovkami, tak bez nich, tj. aby bylo možné program spouštět jak zadáním příkazu `graph.exe x+x^2`, tak příkazem `graph.exe "x + x^2"`. Také použití mezer v zápisu funkce musí být libovolné: Uvědomte si, že funkce se bude analyzátoru předávat jako parametr na příkazové řádce, tj. pokud zápis parametru není uzavřen v uvozovkách, nesmí obsahovat mezery, jinak by byl příkazovým

---

<sup>1</sup>Je třeba, aby bylo možné Váš program přeložit a spustit na PC s operačním prostředím Win32/64 (tj. operační systémy Microsoft Windows NT/2000/XP/Vista/7/8/10/11) a s běžnými distribucemi Linuxu (např. Ubuntu, Debian, Red Hat, atp.). Server, na který budete Vaši práci odevzdávat a který ji otestuje, má nainstalovaný operační systém Debian GNU/Linux 11 (bullseye) s jádrem verze 5.10.0-28-amd64 a s překladačem gcc 10.2.1-6.

procesorem (ve Windows program `cmd`) vyhodnocen jako několik parametrů. Naopak, pokud je nutné do zápisu funkce mezery vložit (z estetických důvodů nebo kvůli přehlednosti), pak musí být zápis funkce uzavřen v uvozovkách, "...". Umožněte proto oba dva způsoby zápisu, tedy „hustý“ zápis bez mezer i zápis s mezerami uzavřený do uvozovek. Je také možné, že uživatel vašeho programu z nepořádnosti někde v zápisu mezery použije a někde zase ne, třeba takto: `"sin (x ^2)*cos(x / 2.0)"` – tato varianta zápisu, ač ošklivá, je také akceptovatelná!

## Způsob zápisu funkcí

V zápisu matematické funkce, jejíž graf bude program generovat, se mohou vyskytnout tyto symboly: (i) konstanty, (ii) proměnná, (iii) aritmetické operátory, (iv) funkce, (v) závorky. Jiné symboly zápis obsahovat nesmí – pokud se v zápisu objeví, měl by program reagovat srozumitelným chybovým hlášením.

Konstanty (i) mohou být ve všech akceptovatelných formátech zápisu celého či reálného čísla v jazyce C, tj. např. i `.5E-02` či `1E0`. S ohledem na to, že má program řešit jen 2D grafy, jediná možná nezávislá proměnná (ii), která se může v zápisu funkce vyskytovat je `x`. V zápisu se nedeklaruje, předpokládáme její implicitní deklaraci (prostě existuje a je připravená k použití uživatelem v zápisu funkce).

Tabulka uvádí přehled všech operátorů (iii), které je možné v zápisu matematické funkce použít:

Operace	Zápis operátoru	Příklad
unární mínus	-	-x
sčítání	+	x + 2
odčítání	-	x - 2
násobení	*	2 * x
dělení	/	x / 2
umocnění	^	x^-2

**Jiné operátory nejsou povolené.** Priorita operátorů je dána matematickými pravidly, případně upravena závorkami ( `a` ) – jiné druhy závorek se v zápisu funkce nesmí vyskytovat. Platný zápis je tedy např.: `(x * sin(x^(2*x))) / 2`.

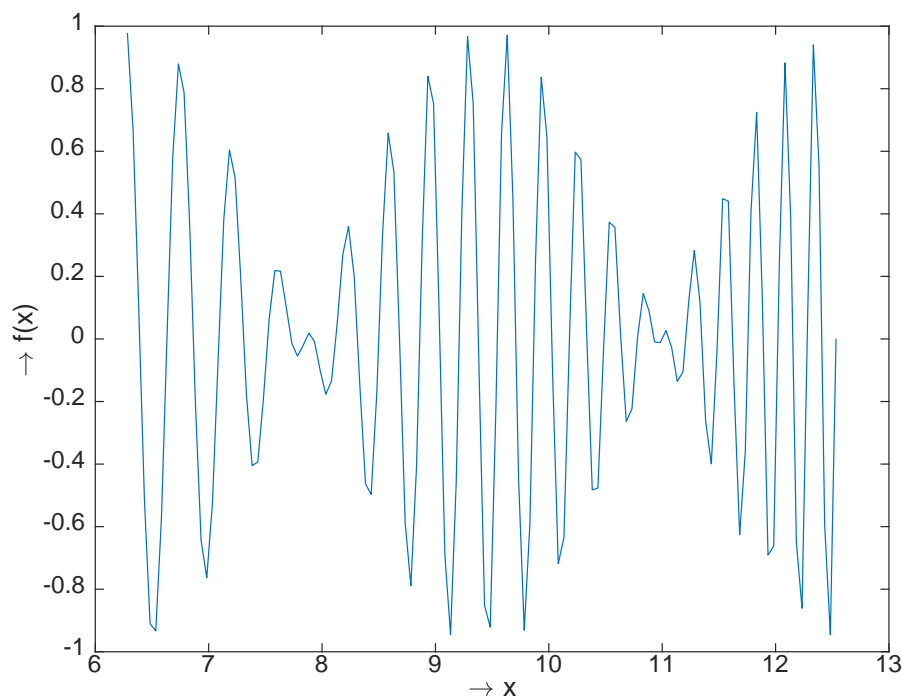
## Specifikace rozsahu zobrazení

Třetí – **nepovinný** – parametr [`<limits>`] na příkazové řádce slouží k předání informací o rozsazích zobrazení grafu analyzované funkce, tedy o horní a dolní mezi definičního oboru a oboru hodnot funkce. Má tvar `<xdol>:<xhor>:<ydol>:<yhor>`. Chceme-li tedy např. zobrazit funkci  $y = \sin(x^2) \cdot \cos(x)$  na intervalu  $\langle 2\pi; 4\pi \rangle$ , přičemž obor hodnot funkce bude roztažen přes celý graf, spustíme program příkazem `graph.exe "sin(x^2)*cos(x)" sin2cos.ps 6.28:12.56:-1:1`. Obrázek 1 ukazuje, jak by měl vypadat výsledek.

Není-li rozsah uveden (protože se jedná o nepovinný parametr), použijte implicitní nastavení, které nechť je pro definiční obor i obor hodnot  $\langle -10; 10 \rangle$ .

## Matematické funkce v zápisu

Váš program by měl akceptovat v zápisu zobrazované funkce některé běžně používané matematické funkce. Funkce uvedené v následujícím výčtu program **musí** akceptovat, jinak nebude uznán za řádně dokončený: absolutní hodnota `abs`; funkce  $e^x$  `exp`, přirozený logaritmus `ln`, dekadický logaritmus `log`; goniometrické funkce `sin`, `cos`, `tan`; cyklometrické funkce `asin`, `acos`, `atan`; hyperbolometrické funkce `sinh`, `cosh`, `tanh`. Další funkce můžete samozřejmě implementovat z vlastní iniciativy.



Obrázek 1: Graf funkce  $y = \sin(x^2) \cdot \cos(x)$  na intervalu  $\langle 2\pi; 4\pi \rangle$ .

## Specifikace výstupu programu

Výstup programu bude směřován na konzoli pouze v případě chyby v zadání parametrů. Pokud budou všechny parametry akceptovatelné a funkce k zobrazení syntakticky správně zapsaná, nemusí program vypisovat na konzoli nic. Pokud program neobdrží požadovaný počet parametrů na příkazovém řádku či pokud je funkce špatně zapsána, měl by vypsát srozumitelné chybové hlášení v angličtině a **skončit nenulovým návratovým kódem**. Návratové kódy jsou určeny tabulkou 1.

Tabulka 1: Návratové kódy programu v případě chyby.

Popis chybového stavu	Návratová hodnota funkce <code>int main()</code>
Bez chyby/korektní ukončení činnosti programu.	0
Nebyly předány správné parametry na příkazové řádce při spuštění programu.	1
Řetězec předaný programu jako první parametr není akceptovatelným zápisem matematické funkce (neobsahuje právě jednu proměnnou $x$ , obsahuje nepovolené operátory, symboly, atp.).	2
Řetězec předaný programu jako druhý parametr nemůže být v hostitelském operačním systému názvem souboru nebo takový soubor nelze vytvořit či do něj ukládat informace.	3
Uvedený nepovinný třetí parametr nelze dekodovat jako rozsahy zobrazení, protože např. obsahuje nepovolené znaky nebo uvedené hodnoty nedávají smysl, apod.	4

Ostatním chybovým stavům můžete přiřadit návratové kódy dle svého uvážení.

Výsledkem činnosti programu by měl být soubor ve formátu PostScript, který bude obsahovat vykreslený graf zadané funkce. PostScript je vlastně zásobníkově orientovaný programovací jazyk podobný jazyku FORTH. Ukázka níže demonstruje základní techniky:

```

%!PS-Adobe-2.0
%%Creator: Kamil
%%Title: untitled
%%CreationDate: Tue Oct 08 10:00:00 2019
%%PageOrder: Ascend
%%EndComments

3 setlinewidth
newpath
0 0 moveto
100 100 lineto
closepath
stroke

.5 setlinewidth
newpath
0 100 moveto
100 0 lineto
closepath
stroke

100 0 lineto
closepath
stroke

0.7 setgray
10 30 moveto
(T) show
0.5 setgray
36 30 moveto
(E) show
0.3 setgray
64 30 moveto
(X) show

showpage

```

Soubory ve formátu PostScript můžete zobrazovat volně dostupným programem GSview, který je grafickým front-endem softwarového R<sup>IP</sup><sup>2</sup> GhostScript – ke stažení z URL <http://www.ghostgum.com.au/software/gsview.htm>.

Úplný popis jazyka PostScript najdete v příručce vydané společností Adobe, která je k dispozici ke stažení na adrese URL <https://www.adobe.com/jp/print/postscript/pdfs/PLRM.pdf>

## Užitečné techniky a odkazy

Uvedené techniky je možné (ale nikoliv nezbytně nutné) využít při řešení úlohy. Protože se jedná o postupy víceméně standardní, lze k nim nalézt velké množství dokumentace:

1. syntaktická analýza rekurzivním sestupem,
2. algoritmus Shunting-yard,
3. zásobník a fronta (a jejich použití pro převod infixového výrazu do polské notace),
4. binární strom.

**Řešení úlohy je zcela ve vaší kompetenci** – zvolte takové algoritmy a techniky, které podle vás nejlépe povedou k cíli. Můžete se volně inspirovat např. webovou aplikací **desmos**, kterou je možno vyzkoušet na adrese URL <https://www.desmos.com/calculator>.

---

<sup>2</sup>Raster Image Processor – software nebo hardware, který převádí jazyk PostScript na bitmapu vhodnou k tisku nebo zobrazení na monitoru.