**Algorithmau a Rhaglenni**

Diffiniad:

Set o reolau diamwys (*unambiguous*) sy’n pennu’r dull (*specify the method*) o ddatrys problem arbennig mewn nifer o gamau meidrol (*finite*).

Mae algorithmau’n bwysig i raglennydd, oherwydd bod rhaid i bob rhaglen gyd‑fynd â nodweddion diffiniad y broblem. Wrth ysgrifennu rhaglen rydym yn rhoi cyfarwyddiadau i beiriant i ddatrys problem, felly rhaid i’r cyfarwyddiadau yma fod yn ddiamwys, a rhaid bod yna derfyn i’r camau sydd eu hangen i ddatrys y broblem.

Dulliau o ddisgrifio algorithmau

Mae yna nifer o ddulliau ar gael. Dyma’r rhai mwyaf cyffredin:

1. “Pseudo-code” (ffug-god)
2. Siart llif rhaglen (*program flowcharts*)
3. Disgrifiad mewn iaith (Cymraeg/Saesneg) strwythuredig (h.y. wedi ei ysgrifennu.)

**“Pseudo-code” (ffug god)**

Defnyddio nodiant sy’n debyg i god iaith rhaglennu, felly mae’n gymharol hawdd i’w drosi i raglen. Gallwch ganolbwyntio ar y rhesymeg (*logic*) heb orfod gofidio am y gystrawen (*syntax*).

Mae’n fwy hyblyg na diagramau strwythur wrth ddod i esbonio’r camau yn fanwl (*detail*). (gweler nodiadau)

1. Marc = 0
2. Canran = 0
3. For i = 0 to 9
4. Mewnbynnu Enw
5. Mewnbynnu Marc
6. Canran = Marc/20 \*100
7. Allbynnu Enw & Canran
8. Next i.

Siart llif rhaglen (*program flowcharts*)

Mae siart llif rhaglen yn dangos y gyfres o gamau i ddatrys y broblem mewn ffurf graffigol. Mae gwahanol siapau yn cynrychioli gwahanol weithredoedd.

Dechrau

Gorffen

Argraffu Enw a Chanran

Loop For M = 1 to 10

Y

M nesaf

N

Darganfod Canran

Mewnbynnu Enw a Marc

Marc = 0

Canran = 0

Disgrifiad mewn iaith (Cymraeg/Saesneg) strwythuredig (h.y. wedi ei ysgrifennu.)

Camau yn cael eu disgrifio trwy ddefnyddio iaith arferol. Gall hyn fod yn llai diamwys na dulliau mwy ffurfiol, ond mae’n gwneud y camau’n haws eu ddeall i gychwyn.

1. Rhoi gwerth cychwynnol o 0 i Marc
2. Rhoi gwerth cychwynnol o 0 i Canran
3. Ar gyfer 10 myfyriwr
4. Mewnbynnu Enw a Marc
5. Cyfrif y Canran
6. Allbynnu Enw a Canran y myfyriwr.
7. Gorffen

**Newidynnau a Chysonion**

# **Newidyn**

Storio gwerth mewn lleoliad gyda dynodwr (ystyrlon). Mae’r gwerth yma’n gallu newid yn ystod rhediad y rhaglen.

# **Cysonyn**

Storio gwerth mewn lleoliad gyda dynodwr (ystyrlon). Nid yw’r gwerth yma’n gallu newid yn ystod y rhediad y rhaglen.

Defnyddiwch gysonyn yn hytrach na’r gwerth llythrennol yn y rhaglen.

## **Pam?**

Mae’n haws i gynnal a chadw’r rhaglen. Os yw’r gwerth yn cael ei ddefnyddio mewn nifer o gyfarwyddiadau’r rhaglen, bydd yn rhaid newid pob un ohonynt os oes rhaid ei newid e.e. newid yn y gyfradd treth ar werth (VAT) o 20% i 15%. Bydd diffyg newid unrhyw un o’r cyfarwyddiadau’n arwain at ganlyniadau anghywir.

Yn gallu gwneud y rhaglen yn fwy dealladwy. Mae fformiwla i amcangyfrif pris gwerthu nwydd, sydd â chyfradd treth o 20% yn fwy dealladwy fel hyn:

PrisGwerthu = CostPrynu + (CostPrynu \* TrethArWerth)

O’i gymharu â:

PrisGwerthu = CostPrynu + (CostPrynu \* 0.2)

Enghraifft o’r defnydd o **Newidynnau** a **Chysonion**

Public Class Form1

'Cysonyn Pi 22/7

Const Pi = 3.14

Private Sub btnArwynebedd\_Click(sender As Object, e As EventArgs) Handles btnArwynebedd.Click

'darganfod arwynebedd Cylch

'Datgan newidynnau

'Arwynebedd

Dim Arwyneddedd As Single = 0.0

'Radiws

Dim Radiws As Integer = 0

'Darllen radiws

Radiws = txtRadiws.Text

'Darganfod arwynebedd

Arwyneddedd = Pi \* (Radiws ^ 2)

'Allbynnu'r canlyniad

lblAteb.Text = "Arwynebedd = " & Arwyneddedd

End Sub

Private Sub btnCylchedd\_Click(sender As Object, e As EventArgs) Handles btnCylchedd.Click

'darganfod arwynebedd Cylch

'Datgan newidynnau

'Cylchedd

Dim Cylchedd As Single = 0.0

'Radiws

Dim Radiws As Integer = 0

'Darllen radiws

Radiws = txtRadiws.Text

'Darganfod cylchedd

Cylchedd = Pi \* Radiws

'Allbynnu'r canlyniad

lblAteb.Text = "Cylchedd = " & Cylchedd

End Sub

End Class

**Dynodwyr (Identifiers)**

Mae **newidyn hollfydol** (global) yn cael ei ddatgelu (*declared*) ar ddechrau’r rhaglen, ac y mae’n cwmpasu (*in scope*) yr holl raglen – h.y. mae’n gallu cael ei ddefnyddio unrhyw le yn y rhaglen.

Mae **newidyn lleol** yn cael ei ddatgelu tu fewn modwl, ac y mae’n cwmpasu’r modwl hynny’n unig – h.y. ond yn gallu cael ei ddefnyddio tu fewn i’r modwl. Nid yw’n bodoli (*exist*) tu allan i’r modwl, ac y mae unrhyw werth oedd yn cael ei storio ganddo’n cael ei golli.

Module Module1

Dim X As Integer

Dim Y As Integer

Sub Main()

X = 2

Y = 3

MsgBox("Gwerth X a Y yn y brif rhaglen cyn galw Newid yw : " & X & " a " & Y)

Newid()

MsgBox("Gerth X a Y yn y brif rhaglen ar ol gadael Newid yw : " & X & " a " & Y)

MsgBox("Gwerth W a Z yn y brif rhaglen yw " & W & " a " & Z)

End Sub

Sub Newid()

Dim W As Integer

Dim Z As Integer

W = 35

X = 6

Y = Y + 1

Z = 3

End Sub

End Module

1. Ar ôl rhedeg y rhaglen beth yw’r allbwn?
2. Pa newidynnau yw’r rhai hollfydol?
3. Pa newidynnau yw’r rhai lleol?
4. Beth ddigwyddodd i werthoedd y newidynnau hollfydol?
5. Beth sy’n digwydd i’r newidynnau lleol o fewn y brif raglen?

Sylw Cyffredinol

Defnyddiwch newidynnau lleol pan fo hyn yn bosibl. Mae hyn yn gwneud y modylau’n fwy annibynnol, sydd â nifer o fanteision. (Gweler uchod)

Paramedrau (*Parameters*)

Paramedr yw’r wybodaeth am eitem o ddata sy’n cael ei chyflenwi (*supplied*) i ffwythiant neu weithdrefn pan mae’r ffwythiant neu weithdrefn yn cael ei alw.

Fel arfer mae paramedrau’n cael eu rhestru tu fewn i gromfachau yn dilyn enw’r ffwythiant neu weithdrefn. Gall paramedr fod yn gysonyn (*constant*), neu’n newidyn (*variable*).

Mae’r llinell isod yn diffinio (defines) Is-reolwaith (Subroutine) yn VB:

Public Sub GwerthDechreuol(ByRef U As Integer ,ByRef I As Integer, ByRef C As Single)

|  |  |
| --- | --- |
| GwerthDechreuol | Dyma enw’r Is-Reolwaith |
| (ByRef U As Integer ,ByRef I As Integer, ByRef C As Single) | Diffinio’r paramedrau a’r math o ddata maent yn mynd i ddal |
| U, I ac C | **Paramedrau Ffurfiol** (Formal Parameters) |

I alw’r is-reolaeth rhaid defnyddio’r llinell isod:

**Call GwerthDechreuol(Uchaf, Isaf, Cyfartaledd)**

|  |  |
| --- | --- |
| Call | Galw is-reolwaith |
| GwerthDechreuol | Enw’r is-reolwaith |
| (Uchaf, Isaf, Cyfartaledd) | Rhestr o ddata sy’n cael eu pasio i’r is-reolwaith – **Paramedrau Gwirioneddol** (Actual Parameters) |

Ymarfer 1

Mae’r isod yn galw a diffinio Ffwythiant (*Function*). Pa un o’r cyfarwyddiadau isod sy’n galw’r ffwythiant, a pha un sy’n diffinio’r ffwythiant?

Labelwch enw’r ffwythiant a pharamedrau’r ffwythiant.

1. If YnEilrif(rhif) Then
2. Public Function YnEilrif(ByVal rhif As Integer) As Boolean

Ymarfer 2

Mae’r isod yn galw a diffinio is-reolwaith. Pa un o’r cyfarwyddiadau is sy’n galw’r is-reolwaith, a pha un sy’n diffinio’r is-reolwaith?

Labelwch enw’r is‑reolwaith a pharamedrau’r is-reolwaith.

1. Call Dwbl (cyntaf, ail)
2. Public Sub Dwbl (ByVal rhif1 As Integer, ByVal rhif2 As integer )

Pam defnyddio paramedrau?

Mae’n gwneud y rhaglen yn haws i’w **deall**, ei **darllen**, ei **datblygu** a’i **gwirio**. Wrth ddefnyddio paramedrau mae’n haws i weld pa ddata, o du allan y modwl, sydd angen ar y modwl, er mwyn iddo gyflawni’r dasg.

Mae’n helpu osgoi newid gwerthoedd newidynnau eang ar ddamwain, a chreu gwallau yn y rhaglen.

Ymarfer 3

Edrychwch ar y darn isod o raglen.

Call Ciwb (rhif, ateb)

Print (“Rhif = “ & rhif & “ Ateb = “ & ateb)

Public Sub Ciwb (ByVal R As Integer ,ByRef A As Integer)

R = R \* R

A = R \* 6

End Sub

Pe bai’r newidynnau â’r gwerthoedd cychwynnol canlynol:

rhif = 2

ateb =0

Beth fydd yw’r allbwn?

Pe bai diffiniad yr is-reolwaith yn newid i:

Public Sub Ciwb (ByVal R As Integer ,ByVal A As Integer)

Beth fydd yr allbwn newydd?

Pe bai diffiniad yr is-reolwaith yn newid i:

Public Sub Ciwb (ByRef R As Integer ,ByRefl A As Integer)

Beth fydd yr allbwn newydd?

**Gweithrediadau mathemategol**

Gweithrediadau mathemategol sylfaenol fel +,-, /, \* defnyddir gyson wrth raglennu. Fodd bynnag, mae gweithrediadau mathemategol eraill y mae angen inni fod yn ymwybodol o'r:

1. **Gweithrediad MOD** : Mae’r gweithredwr hwn yn dychwelyd y gweddill ar ôl rhannu y rhannydd yn rhan annatod o nifer o weithiau y difidend. Gelwir hwn weithiau'n **rhannu gweddill**  E.e. 34 MOD 7 = 6 (h.y. 4 y gweddill 6)

2. Gweithrediad **DIV** (neu weithiau yn dangos fel slaes yn mynd i'r cyfeiriad arall ' \' i'r symbol arferol am rhannu. Dyma pryd rydych yn rhannu dau rif ac yn dychwelyd canlyniad heb y gweddill. Weithiau gelwir hyn yn **rhannu gyfanrif**

e.e. 7 \ 2 = 3

25 \ 4 = 6.

Dangosir hyn fel arfer yn rhaglennu fel 7 DIV 2 = 3 a 25 DIV 4 = 6.

**Ymarfer:**

* Beth yw 38 DIV 7?
* Beth yw 49 DIV 2?
* Beth yw 100 DIV 3?
* Beth yw 166 DIV 12?
* Beth tw 215 DIV 156?
* Beth yw 12 MOD 3?
* Beth yw 51 MOD 9?
* Beth yw 4 MOD 9?
* Beth yw 20 MOD 10?
* Beth yw 102 MOD 6?

Trefnu Data

I osod set o ddata mewn trefn yn ôl rheol bendant e.e.

**DATA DI-DREFN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Carys | Eirian | Alun | Dafydd | Bethan |

**DATA WEDI’U TREFNU’N ESGYNNOL (Ascending Order)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alun | Bethan | Carys | Dafydd | Eirian |

**DATA WEDI’U TREFNU’N DDISGYNNOL (Descending Order)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eirian | Dafydd | Carys | Bethan | Alun |

Mae nifer o ddulliau o drefnu data, sy’n gwahaniaethu o safbwynt:

* Cyflymdra
* Cymhlethdod rhaglennu
* Gofynion storio

Dyma rai ohonynt:

1. **Trefnu Mewnosod (*Insertion*)**
2. **Trefnu Cyfnewid neu Bwrlwm (*Exchange (bubble)*)**
3. **Trefnucyflym (*Quicksort*)**

Nodwch

1. Mae gan rai ieithoedd raglenni (e.e. VB), a’r mwyafrif o gynhyrchwyr rhaglen (*program generators*) ffwythiant safonol sy’n trefnu data. Mae’r rhaglennydd yn defnyddio’r rhain fel Blwch du (black box) – mae’n gwybod beth mae’n gwneud, ond nid sut.
2. Gan fod trefnu data’n gallu bod yn broses araf a chymhleth, mae llawer o sylw wedi cael ei roi i ddatblygu dulliau effeithlon.
3. Mae data’n cael eu htrefnu yng nghof y cyfrifiadur (RAM). Os yw’r data sydd angen eu trefnu ar ffeil ar storfa gynorthwyol, rhaid eu trosglwyddo i’r RAM. Os yw’r RAM yn annigonol i dderbyn yr holl ddata, rhaid eu trefnu mewn darnau, eu danfon i ffeiliau eraill a chyfuno’r ffeiliau.
4. Pam trefnu data?
   * Pan mae rhaid diweddaru data Prif Ffeil gan y data ar Ffeil Trafodion yn ystod swp brosesu, mae’n fwy effeithlon os yw data’r Ffeil Trafod yn yr un drefn â data’r Prif Ffeil.
   * Mae chwilio am ddata’n fwy effeithlon os yw’r data mewn trefn (gweler nodiadau Chwilio)

Dulliau Trefnu

Eto tybiwn ein bod yn defnyddio arae Marciau i storio a’r tro yma trefnu’r data.

Tybiwn, hefyd mae’r rheol fydd i drefnu’r data yn y drefn esgynnol (ascending order)

Cymharu Dulliau Trefnu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dull Trefnu** | **Cyflymdra** | **Cymhlethdod** | Storio | **Cymwysiadau** |
| Swigen  (*Bubble*) | Cymharol araf | Syml, defnyddio dolen | Strwythur data | Nifer fach o eitemau (<=20) |
| Mewnosod  (*Insertion*) | Cyflymach na bybl, ond dal i fod yn araf. | Syml, defnyddio dolen | Strwythur data | Maint isel o ddata (20 - 30) |
| Cyflym (*Quicksort*) | Cyflym | Defnyddio recursion felly er ei fod yn edrych yn syml, angen fwy o feddwl wrth ei ddylunio | Defnyddio recursion, sy’n golygu ei fod yn defnyddio fwy o gof na’r lleill. | Nifer fawr o data (>30) |

1. Trefnu Bwrlwm (Bubble)

Yr enw’n deillio o’r ffaith fod y data’n ymddangos fel eu bod yn arnofio (neu suddo) i’w safle cywir.

Dull

1. Cymharu gwerthoedd y pâr cyntaf o elfennau yn yr arae, h.y. (Marciau(1) a Marciau(2))
2. Os yw Marciau(1) > Marciau(2) aildrefnu’r gwerthoedd (gwerth Marciau(1) yn symud i Marciau(2) a gwerth Marciau(2) yn symud i Marciau(1)
3. Mynd i’r pâr nesaf, sef Marciau(2) a Marciau(3) a’u cymharu, ac aildrefnu os oes angen.
4. Ailadrodd y broses hyd nes bod pob bob un o’r elfennau’n cael eu cymharu. Fydd y gwerth uchaf nawr yn yr elfen olaf o’r arae.
5. Ailadrodd camau 1 i 4 nes bod dim un eitem o ddata wedi newid safle, neu ein bod wedi ailadrodd am (elfennau yn yr arae)-1 o weithiau.

**Dyma’r arae Marciau(6)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *27* | *48* | 13 | 50 | 39 | 42 |

**Cymharu 27 â 48 dim newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 27 | 48 | *13* | 50 | 39 | 42 |

**Cymharu 48 â 13 - newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 27 | 13 | 48 | *50* | 39 | 42 |

**Cymharu 48 â 50 – dim newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 27 | 13 | 48 | *50* | *39* | 42 |

**Cymharu 50 â 39 - newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 27 | 13 | 48 | 39 | *50* | *42* |

**Cymharu 50 â 42 - newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *27* | *13* | 48 | 39 | 42 | 50 |

**Aildrefnu wedi digwydd felly rhaid cymharu eto – pob elfen heb law’r olaf – yn ei safle cywir!**

**Cymharu 27 â 13 - newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 | 27 | 48 | 39 | 42 | ***50*** |

**Cymharu 27 â 48 – dim newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 | 27 | 48 | *39* | 42 | ***50*** |

**Cymharu 48 â 39 - newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 | 27 | 39 | *48* | *42* | ***50*** |

**Cymharu 48 â 42 - newid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 | 27 | 39 | 42 | 48 | ***50*** |

**Aildrefnu wedi digwydd felly rhaid cymharu eto – pob elfen heb law’r 2 olaf – yn ei safle cywir!**

**Cymharu 13 â 27**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 | 27 | 39 | 42 | ***48*** | ***50*** |

**Cymharu 27 â 39**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 | 27 | 39 | 42 | ***48*** | ***50*** |

**Cymharu 39 â 42**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 | 27 | 39 | 42 | ***48*** | ***50*** |

**Dim aildrefnu wedi digwydd felly data yn ei safle cywir!**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***13*** | ***27*** | ***39*** | ***42*** | ***48*** | ***50*** |

Ffug God ar gyfer Trefnu Brwlwm

**repeat  
set a flag to False  
for each pair of keys  
         if the keys are in the wrong order then  
               swap the keys  
               set the flag to True  
         end if  
next pair  
until flag is not set.**

2. Trefnu Mewnosod (Insertion Sort)

* Dewis eitem i’w mewnosod - yr eitem gyntaf yn y rhan o’r rhestr sydd heb ei threfnu.
* Cymharu’r eitem â’r gwerthoedd yn y rhan o’r rhestr sydd wedi ei threfnu, symud yr eitemau yn y rhan yma os oes angen.
* Mewnosod yr eitem yn ei safle cywir yn y rhan o’r arae sydd wedi cael ei threfnu.
* Symud ymlaen i’r eitem nesaf (yr eitem gyntaf yn y rhan o’r rhestr sydd heb ei threfnu).

**Allweddell**

|  |  |
| --- | --- |
|  | Wedi ei trefnu |
|  | Heb ei trefnu |

**Rhestr wreiddiol**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | **5** | 2 | 3 | 1 |

**Gwerth i’w fewnosod = 5**

**Cymharu 4 â 5 – dim newid**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | **5** | 2 | 3 | 1 |

**Cymharu 5 â 2 – symud y 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | **5** | 5 | 3 | 1 |

**Cymharu 4 â 2 – symud y 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | **4** | 5 | 3 | 1 |

**Mewnosod y 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2** | 4 | 5 | 3 | 1 |

**Symud i’r eitem nesaf – 3**

**Cymharu 5 â 3 – symud y 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 4 | **5** | 5 | 1 |

**Cymharu 3 â 4 – symud y 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | **4** | 4 | 5 | 1 |

**Cymharu 2 â 3 – dim newid. Mewnosod y 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | **3** | 4 | 5 | 1 |

**Symud i’r eitem nesaf – 1**

**Cymharu 5 â 1 – symud y 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 3 | 4 | **5** | 5 |

**Cymharu 4 â 1 – symud y 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 3 | **4** | 4 | 5 |

**Cymharu 3 â 1 – symud y 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | **3** | 3 | 4 | 5 |

**Cymharu 2 â 1 – symud y 2.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2** | 2 | 3 | 4 | 5 |

**Mewnosod yr 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | 2 | 3 | 4 | 5 |

**Wedi Trefnu!**

Chwilio Am Ddata

Mae’r broses o chwilio drwy set o ddata er mwyn darganfod gwerth arbennig yn dasg safonol mewn cyfrifiaduro.

* Yn aml mae’r chwilio’n seiliedig ar y maes allweddol.
* Mae yna ddau ganlyniad posibl i chwiliad, naill ai mae’r gwerth yn bresennol a mae’n cael ei ddarganfod, neu nid yw yno a dydym ddim yn ei ddarganfod.

Ystyriwn ddau algorithm chwilio gwahanol, sef:

* Chwiliad Llinol (*Linear Search*)
* Chwiliad Deuaidd (*Binary Search*)

Tybiwn ein bod yn chwilio arae o’r enw Marciau, i ddarganfod y gwerth sy’n cael ei gadw yn y newidyn Targed. Mae’r arae yn storio rhifau cyfan.

Chwiliad Unionlin – Data Di-Drefn (*Unsorted Data*)

* Dyma’r math symlaf o chwiliad.
* Gelwir yn chwiliad unionlin (*linear*), cyfresol (*serial*) neu ddilyniannol (*sequential*).

Sut mae’n gweithio?

Mae’n cychwyn ar ddechrau’r strwythur ac yna’n symud trwyddo yn darllen pob eitem nes ei fod yn darganfod y targed, neu mae’n cyrraedd y diwedd heb ddarganfod y targed, sy’n golygu nad yw’r gwerth yno!

Algorithm:

Tybiwn fod Arae Marciau sydd â “Nifer” o elfennau o fath Rhif Cyfan (*Integer*) wedi ei greu’n barod. Hefyd rydym heb ddangos datgeliadau’r newidyn.

Targed = CInt(txtTarged.Text)

Darganfod = False

Cyfrif = 1

Do While Not (Darganfod) And Cyfrif <= Nifer

If Targed = Marciau(Cyfrif) Then

Darganfod = True

Safle = Cyfrif

Else

Cyfrif = Cyfrif + 1

End If

Loop

If Not (Darganfod) Then

Print ("Nid yw yno")

Else

Print (“Targed yn safle “ & Safle)

End If

Tybiwch fod y data wedi cael eu storio yn yr arae isod.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 22 | 67 | 5 | 34 | 23 | 12 | 7 | 31 | 41 |

Defnyddiwch y tabl isod i ddangos effaith yr algorithm.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nifer | Targed | Cyfrif | Darganfod | Marciau(Cyfrif) | Safle |
| 9 | 23 | 1 | False |  |  |

Nid yw’r Chwiliad Unionlin yn ddull effeithlon o chwilio am ddata. Rhaid chwilio drwy bob elfen cyn darganfod y targed, ac os yw’r targed tuag at ddiwedd yr arae, mae hyn yn gallu cymryd tipyn o amser (yn nhermau amser cyfrifiadurol)!

Os oes 20 elfen mewn arae, ar gyfartaledd, rhaid cymharu 10 elfen cyn dod o hyd i’r targed, (sef, y nifer o elfennau/2,) ac os yw’r targed yn yr elfen olaf, yna mae’n cymryd 20 cymhariaeth.

Mewn sefyllfaoedd lle nad yw’r data mewn trefn y Chwiliad Unionlin yw’r unig ffordd o chwilio!

Defnyddiwch y tabl isod i nodi y nifer o gymariaethau sydd angen eu gwneud ar gyfer darganfod targed mewn arae o’r maint sy’n cael ei roi:

|  |  |  |
| --- | --- | --- |
| Elfennau Mewn Arae | Nifer o Gymariaethau ar Gyfartaledd | Mwyafrif o Gymariaethau |
| 100 |  |  |
| 5000 |  |  |
| 300000 |  |  |

Chwiliad Deuaidd (*Binary Search*) – Data Mewn Trefn (*Sorted Data*)

* Mae’r dull yma o chwilio’n dibynnu ar y ffaith fod y data wedi cael ei osod mewn trefn, (fel arfer yn ôl y maes allweddol).
* Mae’r dull yma’n gyflymach na’r Chwiliad Unionlin, yn enwedig pan fod y nifer o eitemau i’w chwilio’n fawr
* Yn anffodus rhaid sicrhau bod y data mewn trefn, sy’n gallu cymryd amser.
* Hefyd mae’r cod yn fwy cymhleth na chod Chwiliad Unionlin.

Sut mae’n gweithio?

* Mae’n darganfod elfen ganol yr arae. Mae’n cymharu gwerth yr elfen ganol â’r targed.
* Os ydynt yr un peth, yna mae wedi darganfod yr hyn mae i fod.
* Os nad ydynt yr un peth, yna mae’n penderfynu pa ochr o’r arae sy’n rhaid chwilio, drwy gymharu gwerth yr elfen ganol â’r targed:
  + Tybiwn fod y data wedi ei drefnu mewn trefn esgynnol (ascending order).
  + Felly os yw’r targed yn llai na gwerth yr elfen ganol mae’n chwilio’r is-arae ar y chwith;
  + Os yw’r targed yn fwy na’r elfen ganol, yna mae’n chwilio’r is-arae ar y dde.

Mae’n gwneud yn union yr un peth yn yr is-arae, sef:

* + - darganfod yr elfen ganol,
    - cymharu’r data,
    - a naill ai mae wedi darganfod y targed,
    - neu mae’n chwilio yn yr is-arae priodol,
    - hyd nes mae’n darganfod y targed, neu hyd nes ei fod yn ceisio chwilio is-arae gwag, sy’n golygu nad yw’r targed yno.

Yn aml mae tasg yn haws i’w disgrifio trwy ddefnyddio diagram.

Dyma’r Arae Marciau, a’r Targed yw 45. Nodwch fod y data mewn trefn esgynnol.

Cam un - darganfod yr elfen ganol.

**Sut?**

Adio gwerth safle’r elfen cyntaf (1), a gwerth safle’r elfen olaf (9) a defnyddio rhannu rhif cyfan gyda 2. Felly’r elfen ganol yw (1+9)\2 = 10\2 = 5.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 5 | 7 | 12 | 22 | 23 | 45 | 46 | 47 | 50 |

Cam dau - cymharu

Gwerth Marciau(5) yw 23, y targed yw 45, nid ydynt yn hafal (Marciau(5)<>Targed).

Cam tri – ble nesaf?

Mae 45 yn fwy na 23, felly, gan fod yr arae mewn trefn esgynnol, y rhan o’r arae (yr is-arae) sy’n mynd i ddal y targed (os yw yno), yw’r rhan dde, h.y. o elfen 6 i elfen 9.

Felly, y rhan o’r arae rydym am chwilio yw Marciau(6) i Marciau(9):

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | 7 | 8 | 9 |
| 45 | 46 | 47 | 50 |

Cam pedwar – nol i gam un!

Rhaid darganfod elfen ganol yr is-arae yma, sef (6+9)\2 = 15\2 = 7.

Cymharu

Mae Marciau(7) <> Targed - (46<>45)

Ble nesaf?

Gan fod Marciau(7) > Targed - (46>45), mae angen chwilio’r is-arae ar y chwith.

|  |
| --- |
| 6 |
| 45 |

Elfen ganol yw (6+6)\2 =12\2 = 6

Mae’r ddau’n hafal - rydym wedi darganfod y targed.

Er mwyn dod o hyd i’r targed roedd rhaid gwneud 3 cymhariaeth, ond roedd angen ychydig fwy o brosesu, sef darganfod yr elfen ganol. Er mwyn dod o hyd i’r un eitem trwy ddefnyddio Chwiliad Unionlin, byddai angen 6 chymhariaeth.

Beth sy’n digwydd os nad yw’r targed yno?

Pe baem ni heb ddarganfod y targed yn elfen 6 uchod, byddai rhif yr elfen chwith yn uwch na rhif yr elfen dde ar gyfer yr is-arae nesaf. Byddai hyn yn golygu bod y targed ddim yno.

Algorithm:

Mewnbynnu Targed

Darganfod = False

Chwith = 1

Dde = Nifer

Do While Not (Darganfod) And (Dde >= Chwith)

Canol = (Dde + Chwith) \ 2

If Targed = Marciau(Canol) Then

Darganfod = True

Safle = Canol

ElseIf Targed > Marciau(Canol) Then

Chwith = Canol + 1

Else

Dde = Canol - 1

End If

Loop

If (Darganfod = False) Then

Print ("Nid yw yno")

Else

Print(“Safle “ & Safle)

End If

Ymarfer

Defnyddiwch yr arae a’r tabl isod i ddangos beth sy’n digwydd wrth ddefnyddio’r algorithm uchod:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 5 | 7 | 12 | 22 | 23 | 45 | 46 | 47 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nifer | Targed | Darganfod | Chwith | Dde | Canol | Marciau(Canol) |
| 9 | 12 | False |  |  |  |  |
| 9 | 50 | False |  |  |  |  |
| 9 | 21 | False |  |  |  |  |

Datrys Problemau

# **Rhai Technegau Posibl**

1. Rhaid darllen a deall y broblem.
2. Rhannu problem gymhleth (mwy nag un dasg) i is-dasgau mwy syml.
3. Defnyddio profiad
   1. chwilio am dasgau tebyg rydych wedi datrys yn y gorffennol
   2. Addasu/ newid y datrysiadau yma i ddatrys y broblem newydd
4. Ymchwil annibynnol i chwilio am ddatrysiadau ar y we (fforymau ac ati)
5. Gosod cwestiwn ar fforwm addas

# **Problem i’w Ddatrys**

Mewnbynnu data cofrestru dyddiol (5 cyfnod) dosbarth o 10 disgybl . Mae angen darganfod presenoldeb y dosbarth cyfan am yr wythnos, ac yn ddyddiol. Hefyd, mae angen darganfod presenoldeb pob disgybl unigol am yr wythnos, a chwilio am ei bresenoldeb ar gyfer cyfnod penodol.

1. Mewnbynnu data cofrestru 10 disgybl am 5 cyfnod (dydd)
2. Darganfod cyfanswm presenoldeb y dosbarth am yr wythnos
3. Darganfod cyfanswm presenoldeb y dosbarth ar gyfer pob cyfnod (dydd)
4. Darganfod cyfanswm presenoldeb pob disgybl unigol am yr wythnos.
5. Darganfod presenoldeb disgybl penodol ar gyfer cyfnod penodol.

Strwythurau Rheoli (*Control Structures*)

Un o gryfderau cyfrifiadur yw ei allu i weithredu (*execute*) set o gyfarwyddiadau (*instructions*) o fewn rhaglen dro ar ôl tro (*repeat*), a’i allu i ddewis pa gyfarwyddiad i weithredu nesaf. Effaith hyn yw bod gan y rhaglennydd y gallu i reoli’r drefn y mae cyfarwyddiadau yn cael eu gweithredu, yn dibynnu ar y data sy’n cael eu prosesu.

Y cyfarwyddiadau sy’n galluogi’r rhaglennydd i wneud hyn yw **strwythurau rheoli**. Gallwn eu rhannu yn ddau grwp, sef:

* dolennau (*loops*), a
* datganiadau amodol (*conditional statements*).

Heb Strwythurau Rheoli

***Trefn y Gweithredu (Order of Execution)***

Cyfarwyddiad1 1

Cyfarwyddiad2 2

Cyfarwyddiad3 3

Cyfarwyddiad4 4

Cyfarwyddiad5 5

Cyfarwyddiad6 6

Strwythurau Rheoli yn newid trefn y gweithredu

|  |  |  |  |
| --- | --- | --- | --- |
| **Datganiad amodol** (*amodal cyfarwyddiad*) | Trefn y Gweithredu (*Order of Execution*) | **Dolennu** *(Looping)* | **Trefn y Gweithredu** *(Order of Execution)* |
| Cyfarwyddiad1 Cyfarwyddiad2  Cyfarwyddiad3  Cyfarwyddiad4  Cyfarwyddiad5  Cyfarwyddiad6 | 1  2  3  Naid (*Jump*)  4  5 | Cyfarwyddiad1 Cyfarwyddiad2  Cyfarwyddiad3  Cyfarwyddiad4  Cyfarwyddiad5  Cyfarwyddiad6 | 1   1. 6 10 2. 7 11 Ailadrodd x3   4 8 12  5 9 13  14 |

Datganiadau Amodol (*Conditional Statements*)

Mae’r datganiadau yma’n penderfynu pa gyfarwyddiad i weithredu nesaf. Ystyriwn ddau fath, sef:

* If ... Then ... (Else)/ If ... Then ... (ElseIf)
* Select Case

If ... Then ... (Else)/ If ... Then ... (ElseIf)

Ystyriwn y datganiad symlaf yn gyntaf, sef:

If ... Then ...

Mae hyn yn galluogi dethol (*select*) o fewn rhaglen. Mae grwp o gyfarwyddiadau’n gallu cael eu gweithredu, neu’n gallu cael eu hanwybyddu (*ignore*), yn dibynnu ar y data sy’n cael eu prosesu ar y pryd.

|  |  |
| --- | --- |
| **Cystrawen *(Syntax)* Algorithm** | **Enghreifftiau** |
| If <amod> Then | If tymheredd>uchaf Then |
| cyfarwyddiadau | uchaf= tymheredd |
| End if | End if |
|  |  |
|  | If tymheredd >0 Then |
|  | dros=dros+1 |
|  | MsgBox(“Dros Rhewbwynt”) |
|  | End If |

Yn VB medrwch ddefnyddio cystrawen llinell sengl, cystrawen bloc aml-linell (*multiple-line block syntax*):

**If** *amod* **Then** *cyfarwyddiad*

or

**If** *amod* **Then**

*cyfarwyddiad*

**End If**

Fel arfer cymhariaeth (*comparison*) yw’r amod sy’n rhoi gwerth o fath **Boolean**. Mae data o fath Boolean yn gallu bod yn naill ai’n **WIR (*TRUE*)**, neu’n **ANWIR (*FALSE*)**. Os yw’r amod yn **WIR**, mae’r cyfarwyddiadau rhwng y THEN a’r END IF yn **cael eu gweithredu**. Os yw’r amod yn **ANWIR**, mae’r cyfarwyddiadau yma’n **cael eu hanwybyddu**.

Ymarfer 1

Dyma ddarn o gôd ar gyfer darganfod os yw person yn ddigon hen i yrru:

IF (oed < 17) Then

Allbynnu ( “Rhy ifanc i yrru car ar yr heol!”)

End If

Allbynnu(“Hwyl!)

Rhediad Sych (Dry Run)

Defnyddiwch y Tabl Olrhain (*Trace Table*) isod i benderfynu ar yr hyn sy’n cael ei allbynnu gyda’r data profi (*test data*) sydd wedi cael eu darparu.

Tabl Olrhain

|  |  |  |
| --- | --- | --- |
| Oed | Oed<17 (Gwir neu Anwir) | Allbwn |
| 16 |  |  |
| 17 |  |  |
| 18 |  |  |

Ymarfer 2

Dyma darn o gôd sy’n debyg i’r uchod, ond gyda dwy linell o gyfarwyddiadau ar ôl y THEN:

IF (oed < 17) Then

Allbynnu ( “Rhy ifanc i yrru car ar yr heol!”)

Allbynnu (“Dewch nôl pan rydych yn 17!” )

End If

Allbynnu(“Hwyl!)

Rhediad Sych (Dry Run)

Tabl Olrhain

|  |  |  |
| --- | --- | --- |
| Oed | Oed<17 (Gwir neu Anwir) | Allbwn |
| 16 |  |  |
| 17 |  |  |
| 18 |  |  |

**D.S.** Nodwch fod y cyfarwyddiadau o fewn y strwythur rheoli wedi cael eu hindeintio (indented). Mae hyn yn ymarfer da! Mae’n gadael i berson sy’n darllen y rhaglen weld beth sy’n cael ei reoli’n glir. (Hefyd mae marciau’n cael eu rhoi ar ei gyfer!)

Beth yw arwyddocâd y data profi sydd wedi cael eu dewis?

|  |  |
| --- | --- |
| **Oed** | **Arwyddocâd (Beth mae’n profi?)** |
| 16 |  |
| 17 |  |
| 18 |  |

Ymarfer 3

Ewch ati i greu rhaglen yn VB ar gyfer y ddau ymarfer uchod. Profwch fod y rhaglen yn gweithio’n iawn trwy gymharu allbwn eich rhaglen ag allbwn y rhediadau sych. Defnyddiwch yr un data profi ag uchod.

If...Then...Else

Wrth ychwanegu “Else” i If ... Then ... , mae’r rhaglen yn gallu dethol pa un o ddwy set wahanol o gyfarwyddiadau i weithredu, yn dibynnu ar ba ddata sy’n cael eu prosesu. Fe fydd y set arall yn cael ei hanwybyddu.

|  |  |
| --- | --- |
| **Cystrawen *(Syntax)* Algorithm** | **Enghreifftiau** |
| If <amod> Then | If tymheredd >0 Then |
| Cyfarwyddiadau1 | dros=dros+1 |
| else | MsgBox (“Dros Rhewbwynt”) |
| Cyfarwyddiadau 2 | Else |
| End if | is=is+1 |
|  | MsgBox(“Rhewi”) |
|  | End If |

* Os yw’r amod yn **WIR**, mae’r cyfarwyddiadau rhwng y THEN a’r ELSE yn **cael eu gweithredu**, a’r cyfarwyddiadau rhwng yr ELSE a’r END IF yn **cael eu hanwybyddu** .
* Os yw’r amod yn **ANWIR**, mae’r cyfarwyddiadau rhwng y THEN a’r ELSE yn **cael eu hanwybyddu,** cyfarwyddiadau rhwng yr ELSE a’r END IF yn **cael eu gweithredu**.

Ymarfer 4

Medrwn newid y rhaglen gwirio oed gyrrwr nawr i:

IF (oed < 17) Then

Allbynnu ( “Rhy ifanc i yrru car ar yr heol!”)

Allbynnu (“Dewch nôl pan rydych yn 17!” )

Else

Allbynnu(“Rydych yn ddigon hen i yrru”)

Allbynnu((“Danfonwch siec o £30”)

End If

Allbynnu(“Hwyl!)

Rhediad Sych (Dry Run)

Defnyddiwch y Tabl Olrhain (*Trace Table*) isod i benderfynu ar yr hyn sy’n cael ei allbynnu gyda’r data profi (*test data*) sydd wedi cael eu darparu.

Tabl Olrhain

|  |  |  |
| --- | --- | --- |
| Oed | Oed<17 (Gwir neu Anwir) | Allbwn |
| 16 |  |  |
| 17 |  |  |
| 18 |  |  |

Addaswch eich rhaglen VB i gynnwys newidiadau’r ymarfer. Profwch fod y rhaglen yn gweithio’n iawn trwy gymharu allbwn eich rhaglen ag allbwn y rhediadau sych. Defnyddiwch yr un data profi ag uchod.

Cyfuno Amodau – Defnyddio AND, OR a NOT

Mae’n bosibl cyfuno amodau drwy ddefnyddio’r operaduron (operators) AND (AC), OR (NEU) a NOT (NID).

AND (AC)

Rhaid i **bob amod unigol fod yn WIR** er mwyn i’r **amod cyfunol fod yn WIR.** e.e.

IF ( oed > 17 ) and ( oed < 30 ) Then

Allbynnu (“Yn gallu prynu Europass”)

Else

Allbynnu (“Europass ddim ar gael”)

End If

Tabl Olrhain

|  |  |  |  |
| --- | --- | --- | --- |
| **Oed** | **Oed>17** | **Oed<30** | **Allbwn** |
| 16 |  |  |  |
| 17 |  |  |  |
| 24 |  |  |  |
| 30 |  |  |  |
| 31 |  |  |  |

Beth yw arwyddocâd y data profi sydd wedi cael eu dewis?

OR (NEU)

Rhaid i **o leiaf un o’r** **amodau unigol fod yn WIR** er mwyn i’r **amod cyfunol fod yn WIR.** e.e.

IF ( Dydd = “Sadwrn” ) Or ( Dydd = “Sul”) Then

Allbynnu ( “Dim ysgol”)

Else

Allbynnu (“Rhaid mynd i’r ysgol”)

End If

Tabl Olrhain

|  |  |  |  |
| --- | --- | --- | --- |
| **Dydd** | **Dydd = “Sadwrn”** | **Dydd = “Sul”** | **Allbwn** |
| Sadwrn |  |  |  |
| Sul |  |  |  |
| Llun |  |  |  |

Beth yw arwyddocâd y data profi sydd wedi cael eu dewis?

**NOT (NID)**

Rhaid i’r **amod unigol fod yn ANWIR** er mwyn i’r **amod cyfunol fod yn WIR.** e.e.

IF Not( Amser> 8) then

Allbynnu ( “ Rhy gynnar – aros yn y gwely”)

Else

Allbynnu (“ Rhaid codi”)

End If

Tabl Olrhain

|  |  |  |  |
| --- | --- | --- | --- |
| **Amser** | **Amser>8** | **Not(Amser>8)** | **Allbwn** |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |

Beth yw arwyddocâd y data profi sydd wedi cael eu dewis?

Nodyn yngln â VB

Mae yna strwythur arbennig ar gael yn VB sef:

If ... Then ... (ElseIf)

Mae’r strwythur yma’n galluogi’r rhaglen i ddewis rhwng mwy na dwy set o gyfarwyddiadau:

**If** *amod1* **Then**

*cyfarwyddiadau 1*

**ElseIf** *amod N* **Then**

*cyfarwyddiadau N*

**Else**

*cyfarwyddiadau N+1*

**End If**

e.e.

If Dydd = “Sadwrn” Then

Allbynnu (“Chwaraeon heddiw”)

ElseIf Dydd = “Sul” Then

Allbynnu(“Gorffwys heddiw”)

Else

Allbynnu (“Gwaith heddiw”)

End If

Tabl Olrhain

|  |  |  |  |
| --- | --- | --- | --- |
| **Dydd** | **Dydd = “Sadwrn”** | **Dydd = “Sul”** | **Allbwn** |
| Sadwrn |  |  |  |
| Sul |  |  |  |
| Llun |  |  |  |

Beth yw arwyddocâd y data profi sydd wedi cael eu dewis?

Select Case

Mae’r rhaglen yn canghennu (branches) a gweithredu’r cyfarwyddiadau yn dibynnu ar werth y newidyn sy’n rheoli’r cyfarwyddiad Select Case.

Defnydd

Defnyddiwch yn lle If ... Then ... Else, pan fod nifer o bosibiliadau canghennu, a mae’r data o fath trefnnol (ordinal type). Mae’n gwneud y rhaglen yn fwy dealladwy o’i gymharu â defnyddio nifer fawr o If ... Then ...Else.

|  |  |
| --- | --- |
| **Cystrawen *(Syntax)* Algorithm** | **Enghreifftiau** |
| Select Case <newidyn> | Select Case rhiant |
| Case gwerth(oedd)1 | Case “M”,”m” |  |
| cyfarwyddiadau | lblHelo.Text =”Helo Mam” |
| Case gwerth(oedd)2 | Case “T”,”t” |
| cyfarwyddiadau | lblHelo.Text =”Helo Tad” |
| Case Else | End Select |
| cyfarwyddiadau |  |
| End Select |  |

Ymarfer 1

Dyma algorithm sy’n allbynnu un o ddwy neges, yn dibynnu os yw’n benwythnos, neu beidio:

Select Case RhifDydd

Case 1 , 7

Allbynnu (“Hwre! Mae’n benwythnos!”)

Case 2 To 6

Allbynnu (“ Brysiwch! Mae’n amser mynd i’r gwaith “)

Case Else

Allbynnu (“Rhaid i’r rhif bod rhwng 1 a 7”)

End Select

Rhediad Sych (Dry Run)

Defnyddiwch y Tabl Olrhain (*Trace Table*) isod i benderfynu ar yr hyn sy’n cael ei allbynnu gyda’r data profi (*test data*) sydd wedi cael eu darparu.

|  |  |
| --- | --- |
| **RhifDydd** | **Allbwn** |
| 1 |  |
| 7 |  |
| 2 |  |
| 6 |  |
| 5 |  |
| 0 |  |
| 8 |  |

Ymarfer 2

Mae’r algorithm isod yn allbynnu’r nifer o ddyddiau sydd yn y mis cafodd ei fewnbynnu:

1. Mewnbynnu Mis
2. Mewnbynnu Blwyddyn
3. Select case Mis
4. Case 1,3,5,7,8,10,12
   1. Dyddiau=31
5. Case 4,6,9,11
   1. Dyddiau=30
6. Case 2
7. If (Blwyddyn mod 4=0) Then
   1. Dyddiau=29
   2. Else
   3. Dyddiau=28
8. End If
9. End Select
10. Allbynnu Dyddiau

Rhediad Sych (Dry Run)

Defnyddiwch y Tabl Olrhain (*Trace Table*) isod i benderfynu ar yr hyn sy’n cael ei allbynnu gyda’r data profi (*test data*) sydd wedi cael eu darparu.

|  |  |  |  |
| --- | --- | --- | --- |
| Mis | Blwyddyn | Dyddiau | Allbwn |
| 1 | 2003 |  |  |
| 8 | 2003 |  |  |
| 4 | 2003 |  |  |
| 11 | 2003 |  |  |
| 12 | 2003 |  |  |
| 2 | 2003 |  |  |
| 2 | 2006 |  |  |
| 2 | 2004 |  |  |

Ymarfer 3

Ewch ati i greu rhaglen yn VB ar gyfer y ddau ymarfer uchod. Profwch fod y rhaglen yn gweithio’n iawn trwy gymharu allbwn eich rhaglen ag allbwn y rhediadau sych. Defnyddiwch yr un data profi ag uchod.

Dolennau (*Loops*)

Yma mae’r rhaglen yn ailwneud set o gyfarwyddiadau. Fel arfer mae yna derfyn i’r nifer o weithiau mae’r cyfarwyddiadau’n cael eu hailadrodd. Heb hyn fe fydd y ddolen yn ddiderfyn (*perpetual loop*), a rhaid crasio allan o’r rhaglen er mwyn ei orffen.

Mae yna nifer o ffyrdd o reoli dolen, ond medrwn eu rhannu i dri chategori, sef:

1. Rhai sy’n profi’r amod rheoli cyn dechrau gweithredu’r cyfarwyddiadau.
2. Rhai sy’n profi’r amod rheoli ar ddiwedd gweithredu’r cyfarwyddiadau.
3. Rhai sy’n gweithredu’r cyfarwyddiadau nifer osodedig o weithiau.

Amodau Rheoli (Control Conditions)

Mae’r ddwy ffordd sy’n cael eu rhestru’n gyntaf uchod yn defnyddio amod rheoli. Mae yna ddau fath ar gael, sef:

1. Rhai sy’n gwneud y ddolen tra bod yr amod yn WIR – dolennau WHILE.
2. Rhai sy’n gwneud y ddolen tra bod yr amod yn ANWIR – dolennau REPEAT .. UNTIL.

Profi’r Amod Rheoli Cyn Dechrau Gweithredu’r Cyfarwyddiadau (Do While)

Tra bod yr amod(au)’n WIR mae’r rhaglen yn gwneud cyfarwyddiadau’r ddolen.

|  |  |
| --- | --- |
| **Cystrawen *(Syntax)* Algorithm** | **Enghreifftiau** |
| Do While <condition> | Do While swm<1000 |
| statements | swm = swm\* 1.1 |
| Loop | Loop |

|  |  |
| --- | --- |
|  | Do While (mwy<>”Y”) OR (mwy<>”N”) |
|  | MsgBox(“Rhaid dewis Y neu N”) |
|  | Mwy = InputBox(“Rhaid dewis Y neu N”) |
|  | Loop |

Ymarfer 1

Mae’r rhaglen isod yn darganfod cyflog 7 person. Rhaid mewnbynnu enw ac oriau pob gweithiwr.

gweithwyr = 0

Do while (gweithwyr < 7)

Enw=InputBox(“Enw’r gweithiwr?”)

Oriau= InputBox(“oriau gweithiwyd? “)

Cyflog = oriau \* 5.50

Allbynnu ( “Cyflog yw “ & Cyflog)

gweithwyr = gweithwyr +1

Loop

Allbynnu (“Gweithiwr wedi eu talu!“)

Rhediad Sych (Dry Run)

Defnyddiwch y Tabl Olrhain (*Trace Table*) isod i benderfynu ar yr hyn sy’n cael ei allbynnu gyda’r data profi (*test data*) sydd wedi cael eu darparu.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Enw** | **Oriau** | **Cyflog** | **Gweithwyr** | **Allbwn** |
| Ioan | 40 |  |  |  |
| Lisa | 39 |  |  |  |
| Dafydd | 41 |  |  |  |
| Sioned | 29 |  |  |  |
| Sian | 55 |  |  |  |
| Huw | 30 |  |  |  |
| Ceri | 40 |  |  |  |

Ymarfer 2

Golygwch y rhaglen uchod i ddarganfod y cyfanswm mae’n rhaid i’r cwmni ei dalu mewn cyflogau.

gweithwyr = 0

Cyfanswm = 0

Do while (gweithwyr < 7)

Enw=InputBox(“Enw’r gweithiwr?”)

Oriau= InputBox(“oriau gweithiwyd? “)

Cyflog = oriau \* 5.50

Allbynnu ( “Cyflog yw “ & Cyflog)

**Cyfanswm = Cyfanswm + Cyflog**

gweithwyr = gweithwyr +1

Loop

Allbynnu (“Gweithiwr wedi eu talu!“)

Allbynnu (“ Cyfanswm cyflog yw “ & Cyfanswm)

Rhediad Sych (Dry Run)

Defnyddiwch y Tabl Olrhain (*Trace Table*) isod i benderfynu ar yr hyn sy’n cael ei allbynnu gyda’r data profi (*test data*) sydd wedi cael eu darparu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Enw** | **Oriau** | **Cyflog** | **Gweithwyr** | **Cyfanswm** | **Allbwn** |
| Ioan | 40 |  |  |  |  |
| Lisa | 39 |  |  |  |  |
| Dafydd | 41 |  |  |  |  |
| Sioned | 29 |  |  |  |  |
| Sian | 55 |  |  |  |  |
| Huw | 30 |  |  |  |  |
| Ceri | 40 |  |  |  |  |

Ymarfer 3

Ewch ati i greu rhaglen yn VB ar gyfer y ddau ymarfer uchod. Profwch fod y rhaglen yn gweithio’n iawn trwy gymharu allbwn eich rhaglen ag allbwn y rhediadau sych. Defnyddiwch yr un data profi ag uchod.

Ymarfer 4 (Dewisol)

Ysgrifennwch raglen i fewnbynnu enw a marc canrannol 5 disgybl. Rhaid allbynnu enw a gradd pob disgybl, a’r marc cyfartaledd.

Ymarfer 5

Yn yr enghreifftiau uchod rydym yn gwybod yn union y nifer o weithiau i ailwneud y cyfarwyddiadau. Beth am sefyllfaoedd pan nad ydym yn gwybod o flaen llaw y nifer o weithiau i ailadrodd? Yn hytrach na defnyddio cyfrif (*count*) rhaid defnyddio dull arall.

Mae’r algorithm isod yn darganfod y nifer o flynyddoedd y byddai’n cymryd i swm sy’n cael ei fewnbynnu cyrraedd miliwn o bunnoedd, gyda chyfradd llog o 10%.

1. Mewnbynnu swm
2. Blynyddoedd=0
3. Do While swm>1000000
4. swm = swm \* 1.1
5. Blynyddoedd = Blynyddoedd + 1
6. Loop
7. Allbynnu (“Nifer o flynyddoedd yw “ & Blynyddoedd)

Ysgrifennwch raglen ar gyfer hyn..

Ymarfer 6

Ambell waith rhaid mewnbynnu cyfres o ddata, sy’n gallu amrywio mewn nifer e.e. darganfod cyfanswm cost nwyddau mewn siop. Mae’r nifer o nwyddau sy’n cael eu harchebu’n amrywio gyda phob cwsmer. Y tro yma rhaid defnyddio gwerth gwyliedydd (*sentinel value*).

1. Nifer = 0
2. Cyfanswm = 0
3. Pris = InputBox (“Pris”)
4. Do While Pris<>-1
5. ‘ Ychwanegu un i cyfanswm y nwyddau
6. Nifer = Nifer + 1
7. ‘ Ychwanegu’r pris i’r cyfanswm
8. Cyfanswm = Cyfanswm + Pris
9. Pris = InputBox (“Pris”)
10. Loop
11. Select Case Nifer
12. Case 0
13. Allbynnu (“Cafodd ddim ei brynnu”)
14. Case 1
15. Allbynnu (“ Cyfanswm £“ & Cyfanswm)
16. Allbynnu (“Cafodd un eitem ei brynnu”)
17. Case Else
18. Allbynnu (“ Cyfanswm £“ & Cyfanswm)
19. Allbynnu(“Cafodd “ & Nifer & “eitemau eu brynnu”)
20. End Select

Rhediad Sych (Dry Run)

Defnyddiwch y Tabl Olrhain (*Trace Table*) isod i benderfynu ar yr hyn sy’n cael ei allbynnu gyda’r data profi (*test data*) sydd wedi cael eu darparu.

**Tabl Un**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nifer** | **Cyfanswm** | **Pris** | **Allbwn** |
|  |  | 2.30 |  |
|  |  | 0.55 |  |
|  |  | 1.10 |  |
|  |  | -1 |  |

**Tabl Dau**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nifer** | **Cyfanswm** | **Pris** | **Allbwn** |
|  |  | 4.10 |  |
|  |  | -1 |  |

**Tabl Tri**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nifer** | **Cyfanswm** | **Pris** | **Allbwn** |
|  |  | -1 |  |

**Tabl Un**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nifer** | **Cyfanswm** | **Pris** | **Allbwn** |
|  |  | 1.30 |  |
|  |  | 0.75 |  |
|  |  | 0.55 |  |
|  |  | 1.00 |  |
|  |  | -1 |  |

Ysgrifennwch raglen ar gyfer hyn..

Rhai sy’n profi’r amod rheoli ar ddiwedd gweithredu’r cyfarwyddiadau. (Do Until)

Yma mae’r cyfarwyddiadau’n cael ei wneud o leiaf unwaith cyn i’r amod cael ei wirio.

|  |  |
| --- | --- |
| **Cystrawen *(Syntax)* Algorithm** | **Enghreifftiau** |
| Loop (Do) | Loop (Do) |
| Cyfarwyddiadau | MsgBox (“Dyfalwch nawr) |
| Until <amod> | Dyfaliad=InpuBox(“Dyfalwch”) |
| Loopend | Until dyfaliad=rhif |
|  | Loopend |

Ymarfer 1

Mae’r algorithm isod yn cynhyrchu haprif (*random number*) rhwng 1 a 10. Rhaid i’r defnyddiwr dyfalu’r rhif. Ysgrifennwch y rhaglen yn VB.

1. rhif =random(10)
2. Do
3. mewnbynnu (dyfaliad) ‘Defnyddiwch InputBox
4. Until dyfaliad = rhif
5. Allbynnu (“Cywir!)

Ymarfer 2

Golygwch y rhaglen i gynnwys cyfrif (*count*) i rifo’r nifer o ddyfaliadau.

Ymarfer 3

Golygwch y rhaglen i allbynnu’r neges “Anghywir” ar gyfer dyfaliadau anghywir.

Ymarfer 4

Golygwch y rhaglen i helpu’r defnyddiwr wrth ddweud wrtho os yw’r dyfaliad yn uwch neu’n is na’r haprif.

Dolen Sy’n Cael ei Gweithredu Nifer Osodedig o Weithiau ( For ... Next)

Pan rydych am reoli dolen am nifer osodedig o weithiau, defnyddiwch For ... Next. Mae’n defnyddio cownter i reoli’r nifer o weithiau i ailadrodd y cyfarwyddiadau.

|  |  |
| --- | --- |
| **Cystrawen *(Syntax)* Algorithm** | **Enghreifftiau** |
| Loop For <condition> Step <camau> | Loop For cymeriad='a' to 'z' |
| cyfarwyddiadau | Allbynnu (cymeriad) |
| Loopend | Loopend |
|  |  |
|  | Loop For lluos= 1 to 12 |
|  | ateb=lluos \* 2 |
|  | allbynnu ( lluos & “\* 4 =” & ateb) |
|  | Loopend |

Y gystrawen yn VB yw:

**For** *cyfrif* = *dechrau* **To** *diwedd* [**Step** *camau*]

*cyfarwyddiadau*

**Next** [*cyfrif*]

Ymarfer 1

Dyma raglen sy’n amcangyfrif y tablau.

1. Mewnbynnu rhif
2. For cyfrif = 1 to 12
3. ateb = cyfrif \* rhif
4. Allbynnu (cyfrif & “ \* “ & rhif & “ = “ & ateb)
5. Next

Ymarfer 2

Newidiwch linell 2 i:

2. For cyfrif = 12 to 1 Step –1.

Nodwch y gwahaniaeth.

Ymarfer 3

Mae’r rhaglen isod yn allbynnu cymeriadau, gyda gwerth ASCII o 32 i 255.

Dim Cymeriad As Integer

frm\_ascii.WindowState = 2

For Cymeriad = 0 To 255

lst\_allbwn.AddItem (Cymeriad & " " & Chr(Cymeriad))

Next

Ymarfer 4

Mae’r rhaglen isod yn tynnu patrwm ar y Ffurflen (Form).

Dim X As Integer

Dim Y As Integer

Dim r, g, b As Integer

frm\_Patrwm.Show

For X = 0 To ScaleWidth Step 50

Randomize

r = Int(Rnd() \* 255 + 1)

g = Int(Rnd() \* 255 + 1)

b = Int(Rnd() \* 255 + 1)

Line (X, 0)-(ScaleWidth - X, ScaleHeight), RGB(r, g, b)

Next X

For Y = 0 To ScaleHeight Step 50

Randomize

r = Int(Rnd() \* 255 + 1)

g = Int(Rnd() \* 255 + 1)

b = Int(Rnd() \* 255 + 1)

Line (0, Y)-(ScaleWidth, ScaleHeight - Y), RGB(r, g, b)

Next Y

Newidiwch werth y cam o 50 i 100 i 200. Beth sy’n digwydd?

Ymarfer 5

Mae yna 5 beirniad yn marcio cystadleuaeth sglefrio (*skateing competition*). Mae cyfanswm y marciau’n cael ei ddarganfod, ond mae’r marc isaf a’r uchaf yn cael eu hanwybyddu. Dyma raglen ar gyfer gwneud y dasg. Nodwch fod un ddolen wedi cael ei nythu o fewn dolen arall.

Dim Uchaf As Single

Dim Isaf As Single

Dim Cyfanswm As Single

Dim Marc As Single

Dim Beirniad As Integer

Dim Dilys As Boolean

'Gwerthoedd cychwynnol

frm\_Sglefrio.Show

Uchaf = -1

Isaf = 7

Cyfanswm = 0

'Dechrau'r ddolen

For Beirniad = 1 To 5

Dilys = True

Do

'mewnbynnu marc

Marc = InputBox("Marc Beirniad " & Beirniad,"Marciau")

'dilysu'r marc

If (Marc < 0) Or (Marc > 6) Then

MsgBox ("Marc annilys, rhaid bod o 0 i 6")

Dilys = False

Else

Dilys = True

End If

Loop Until Dilys

'Ychwanegu'r marc i'r cyfanswm

Cyfanswm = Cyfanswm + Marc

'Dileu'r marc isaf

If Marc < Isaf Then

Isaf = Marc

End If

'Dileu'r marc uchaf

If Marc > Uchaf Then

Uchaf = Marc

End If

Next

'Allbynnu'r cyfanswm

Cyfanswm = Cyfanswm - (Isaf + Uchaf)

lblCyfanswm.Text = ("Cyfanswm marciau'r sglefriwr " & Cyfanswm)

Rhedwch y rhaglen, gan ddefnyddio’r data isod:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1. Normal | 1. Anarferol 1 | 1. Annormal 2 | 1. Annormal 3 | 1. Annilys |
|  | 4.9 | 3.2 | 0 | 6 | -1 (4) |
|  | 4.6 | 4.8 | 0 | 6 | 6.1 (6) |
|  | 5.0 | 3.2 | 0 | 6 | 7 (-1) 5 |
|  | 4.8 | 4.8 | 0 | 6 | -1 (7) (8) 5 |
|  | 4.7 | 4.0 | 0 | 6 | -0.1 (0.1) |
| Cyfanswm |  |  |  |  |  |

Ymarfer 6

Addaswch y rhaglen i ddelio gyda 10 cystadleuydd.

Beth oedd rhaid gwneud?

Rhaglennu Modylol (*Modular Programming*)

Nodweddion Rhaglen Dda

Dyma restr o nodweddion rhaglen dda:

* dibynadwy (*dependable*) - rhaid i’r rhaglen gwneud beth y dylai bob tro;
* hawdd i gynnal (*maintain*) - rhaid medru ei newid os oes angen;
* strwythuredig (*structured*) – hawdd i weld pa ran sy’n gwneud pa dasg;
* dealladwy (*understandable*) - rhaid deall y rhaglen er mwyn ei chynnal;
* darllenadwy (*readability*) - rhaid i raglenwyr eraill ddeall y rhaglen er mwyn ei chynnal;
* effeithlonrwydd (*efficiency*) - rhaid i’r rhaglen wneud y dasg yn gyflym ac yn effeithlon
* arbed cof (*memory*) - dylai’r rhaglen gymryd cyn lleied o le yn y cof ag sy’n bosib, er gyda phris cof mor rhad heddiw nid yw hyn mor bwysig.

Beth yw Rhaglennu Modylol?

Roedd rhaglenni cynnar yn tueddu i fod yn fawr a heb strwythur (*structure*), ac o ganlyniad nid oeddent yn cwrdd â nifer o’r nodweddion uchod. Felly roedd y rhaglenni yma’n anodd ei darllen a’u deall, ac o ganlyniad yn anodd ei chynnal.

Cynigodd Rhaglennu Modylol ddatrysiad i’r broblem, trwy gynnig dull a oedd yn rhannu’r prif raglen i nifer o is-dasgau, neu fodylau (*modules*).

Mae gan raglennu modwlar dri syniad sylfaenol (*basic idea*), sef:

* Mae rhaglen yn cael ei rhannu i nifer o fodylau annibynnol, sy’n gallu cael eu casglu (*complie*) ar wahân.
* Dylai pob modwl fod yn gymharol fach (50 i 100 o linellau ar y mwyaf.
* Dylai pob modwl gael eu profi’n annibynnol cyn eu hintegreiddio i’r brif raglen.

Egwyddorion Rhaglen Modylol Dda

Dylid defnyddio’r Dull Top-i-Lawr (*Top-Down Method)* a’r Dull Coethder Camau (*Step-Wise Refinement*), er mwyn creu rhaglen sy’n cwrdd â’r egwyddorion uchod. Dylai’r modylau:

* Gwneud tasg benodol;
* Heb fod yn rhy hir;
* Mor annibynnol (*self-contained*) ag sy’n bosibl.

Manteision Rhaglennu Modylol

* 1. Ar ôl creu’r modwl, gallwch ei ddefnyddio unrhyw nifer o weithiau, naill ai o fewn yr un rhaglen, neu mewn rhaglenni eraill. Diffinio unwaith – defnyddio cymaint o weithiau rydych yn dymuno.
  2. Gan fod modwl yn cynnwys nifer cymharol fach o linellau cod, dylai fod yn hawdd i’w:
  + ddarllen;
  + ddeall;
  + gywiro;

sy’n golygu y bydd yn fwy dibynadwy.

* 1. Mae’n haws i gynnal y rhaglen, oherwydd gallwn ddarganfod a newid y modwl sy’n cael ei effeithio'n gyflym.
  2. Trwy ddefnyddio’r Dyluniad Top-i-Lawr, gall y rhaglen ddilyn y dyluniad yn hawdd.
  3. Wrth greu rhaglenni mawr, mae tîm o raglenwyr yn cael ei ddefnyddio. Gall pob rhaglennydd dderbyn un, neu grŵp o fodylau, i’w creu. Mae hyn yn galluogi modylau gael eu datblygu yn baralel, sy’n cyflymu datblygiad y rhaglen.
  4. Wrth ddefnyddio tîm o raglenwyr, gall y rhaglenwyr mwyaf profiadol ddatblygu’r modylau mwyaf cymhleth, a’r rhai llai profiadol ddatblygu’r rhai llai cymhleth.
  5. Gall modylau gael eu profi’n annibynnol. Gall hyn gwtogi ar yr amser mae’n cymryd i gael y rhaglen i weithio, ac mae’n haws cywiro gwallau.
  6. Pe bai rhaglennydd yn gadael yn ystod datblygiad y rhaglen, mae’n haws i’r rhaglennydd newydd ymgymryd â’r gwaith o orffen modwl yn hytrach na rhaglen gyfan.
  7. Mae’n haws i fonitro a rheoli rhaglen fawr.

Anfanteision Rhaglennu Modylol

1. Mae’n aneffeithlon (*inefficient*) o safbwynt y cyfrifiadur. Mae galw modylau yn golygu bod y cyfrifiadur yn gorfod nodi pa fodwl sy’n cael ei alw, ble cafodd ei alw o fewn y rhaglen, pa baramedrau gafodd eu defnyddio ac ati – tasgau sy’n arafu’r gwaith o brosesu.
2. Mae cydweithio mewn tîm yn gallu bod yn gymhleth, gyda chyfarfodydd i sicrhau cysondeb ac ati
3. Rhaid dibynnu ar eraill i orffen eu tasgau.

Rhaglennu Modylol yn VB

* + Rhaid rhoi dynodwr (*identifier*) i’r modwl.
  + Mae gan fodwl yr un strwythur sylfaenol (*basic structure*) â rhaglen.
  + I alw (call) modwl rydych yn defnyddio ei ddynodwr, neu’n aml yn VB, rhaid gosod y gair “Call” o flaen y dynodwr e.e. i alw modwl o’r enw Mewnbwn:
  + Call Mewnbwn
  + Er i’r modwl orffen cyflawni’r cyfarwyddiadau (instructions) mae’n dychwelyd i gyfarwyddyd yn syth ar ôl y man y cafodd ei alw.
  + Rhaid bod y modwl wedi cael ei greu cyn y gellir ei alw.

Rhaid gwahaniaethu rhwng dau fath o fodwl, sef:

**Gweithdrefn (*Procedure*) a Ffwythiant (*Function*).**

Maent yn debyg mewn nifer o agweddau, ond mae yna wahaniaethau rhyngddynt.

Y gwahaniaeth rhwng Gweithdrefn a Ffwythiant

Mae **ffwythiant** bob tro’n **dychwelyd gwerth (*returns a* *value***), trwy drosglwyddo gwerth iddo’i hun o fewn y ffwythiant. **Rhaid i’r gwerth fod o fath data syml** (*simple data type*), **nid ydynt yn medru dychwelyd mwy nag un gwerth**, ac nid ydynt yn gallu dychwelyd strwythurau data (*data structures*).

e.e. Dyma raglen ar gyfer ffwythiant sy’n penderfynu os yw rhif yn eilrif (*even number*) neu’n odrif (*odd number*):

‘rhan o’r brif raglen sy’n galw’r ffwythiant a delio gyda’r gwerth sy’n cael ei ddychwelyd

‘ mae’r ffwythiant yn dychwelyd gwerth Boolean (*Boolean value*)

Private Sub cmd\_Penderfynu\_Click()

Dim rhif As Integer

rhif = txt\_Rhif.Text

If YnEilrif(rhif) Then

lbl\_Rhif.Caption = "Mae'n Eilrif"

Else

lbl\_Rhif.Caption = "Mae'n Odrif"

End If

End Sub

‘Algorithm ar gyfer y ffwythiant YnEilrif, sy’n dychwelyd gwerth Boolean gydag un paramedr, sef rhif sy’n gyfanrif (*integer*), ac un newidyn lleol (*local variable*) sef temp.

Function yneilrif (ByVal rhif As Integer) As Boolean

Public Function YnEilrif(ByVal rhif As Integer) As Boolean

Dim temp As Integer

'mae \ yn golygu rhannu cyfanrif (*integer division*)

temp = rhif \ 2

temp = temp \* 2

'os yw'r canlyniad yr un gwerth yna mae'r rhif yn eilrif

If temp = rhif Then

YnEilrif = True

Else

YnEilrif = False

End If

End Function

**Tabl olrhain (Trace Table)** ar gyfer Ffwythiant yneilrif, gyda un eilrif (4) ac un odrif (5)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cam (*Step*) | yneilrif | rhif | temp | yneilrif | rhif | temp |
| 1 |  | 4 |  |  | 5 |  |
| 3 |  | 4 | 2 |  | 5 | 2 |
| 4 |  | 4 | 4 |  | 5 | 4 |
| 6 |  | 4 | 4 |  | 5 | 4 |
| 7 | True | 4 | 4 |  |  |  |
| 9 |  |  |  | False | 5 | 4 |
| 11 | True | 4 | 4 | False | 5 | 4 |

Gyda rhif=4, allbwn y brif raglen:

Mae 4 yn eilrif

Gyda rhif=5, allbwn y brif raglen:

Mae 5 yn odrif

**Nid yw gweithdrefn yn dychwelyd gwerth**. Mae’n **gallu newid gwerth ei baramedrau** a chadw’r newidiadau yma ar gyfer y brif raglen, ac y mae’r **paramedrau yma’n gallu bod o unrhyw fath**, mathau data syml (*simple data types*), neu strwythurau data (*data structures*) ond nid oes rhaid. (Gweler nodiadau isod).

Cwmpas Dynodwyr – Newidyn Eang a Newidyn Lleol

*(Scope of Identifiers - Global Variables and Local variables)*

Mae newidyn hollfydol yn cael ei ddatgelu (*declared*) ar ddechrau’r rhaglen, ac y mae’n cwmpasu (*in scope*) yr holl raglen – h.y. mae’n gallu cael ei ddefnyddio unrhyw le yn y rhaglen.

Mae newidyn lleol yn cael ei ddatgelu tu fewn modwl, ac y mae’n cwmpasu’r modwl hynny’n unig – h.y. ond yn gallu cael ei ddefnyddio tu fewn i’r modwl. Nid yw’n bodoli (*exist*) tu allan i’r modwl, ac y mae unrhyw werth oedd yn cael ei storio ganddo’n cael ei golli.

Er mwyn deall yn well ystyr cwmpasu, teipiwch y rhaglen ganlynol i mewn i’r cyfrifiadur, ac atebwch y cwestiynau sy’n dilyn.

Option Explicit

Dim X As Integer

Dim Y As Integer

Private Sub Form\_Load()

frm\_HollfydolALleol.Show

'gwerth dechreuol i X and Y

X = 2

Y = 3

Print ("Gwerth X ac Y yn y brif raglen cyn galw Newid yw " & X & " " & Y & vbCrLf)

Call Newid

Print ("Gwerth X ac Y yn y brif raglen ar ôl gadael Newid yw " & X & " " & Y & vbCrLf)

Print ("Gwerth W Z yn y brif raglen yw " & W & " " & Z & vbCrLf)

End Sub

Public Sub Newid()

Dim W As Integer

Dim Z As Integer

W = 35

X = 6

Y = Y + 1

Z = 3

Print ("Gwerth X Y W Z tu fewn Newid yw " & X & " " & Y & " " & W & " " & Z & vbCrLf)

End Sub

1. Ar ôl rhedeg y rhaglen beth yw’r allbwn?

2. Pa newidynnau yw’r rhai hollfydol?

1. Pa newidynnau yw’r rhai lleol?
2. Beth ddigwyddodd i werthoedd y newidynnau hollfydol?
3. Beth sy’n digwydd i’r newidynnau lleol o fewn y brif raglen?

Sylw Cyffredinol

Defnyddiwch newidynnau lleol pan fo hyn yn bosibl. Mae hyn yn gwneud y modylau’n fwy annibynnol, sydd â nifer o fanteision. (Gweler uchod)

Paramedrau (*Parameters*)

Paramedr yw’r wybodaeth am eitem o ddata sy’n cael ei chyflenwi (*supplied*) i ffwythiant neu weithdrefn pan mae’r ffwythiant neu weithdrefn yn cael ei alw.

Fel arfer mae paramedrau’n cael eu rhestru tu fewn i gromfachau yn dilyn enw’r ffwythiant neu weithdrefn. Gall paramedr fod yn gysonyn (*constant*), neu’n newidyn (*variable*).

Mae’r llinell isod yn diffinio (defines) Is-Reolwaith (Subroutine) yn VB:

Public Sub GwerthDechreuol(ByRef U As Integer ,ByRef I As Integer, ByRef C As Single)

|  |  |
| --- | --- |
| GwerthDechreuol | Dyma enw’r Is-Reolwaith |
| (ByRef U As Integer ,ByRef I As Integer, ByRef C As Single) | Diffinio’r paramedrau a’r math o ddata maent yn mynd i ddal |
| U, I ac C | **Paramedrau Ffurfiol** (Formal Parameters) |

I alw’r is-reolaeth rhaid defnyddio’r llinell isod:

**Call GwerthDechreuol(Uchaf, Isaf, Cyfartaledd)**

|  |  |
| --- | --- |
| Call | Galw is-reolwaith |
| GwerthDechreuol | Enw’r is-reolwaith |
| (Uchaf, Isaf, Cyfartaledd) | Rhestr o ddata sy’n cael eu pasio i’r is-reolwaith – **Paramedrau Gwirioneddol** (Actual Parameters) |

Ymarfer 1

Mae’r isod yn galw a diffinio Ffwythiant (*Function*). Pa un o’r cyfarwyddiadau isod sy’n galw’r ffwythiant, a pha un sy’n diffinio’r ffwythiant?

Labelwch enw’r ffwythiant a pharamedrau’r ffwythiant.

1. If YnEilrif(rhif) Then
2. Public Function YnEilrif(ByVal rhif As Integer) As Boolean

Ymarfer 2

Mae’r isod yn galw a diffinio is-reolwaith. Pa un o’r cyfarwyddiadau is sy’n galw’r is-reolwaith, a pha un sy’n diffinio’r is-reolwaith?

Labelwch enw’r is‑reolwaith a pharamedrau’r is-reolwaith.

1. Call Dwbl (cyntaf, ail)
2. Public Sub Dwbl (ByVal rhif1 As Integer, ByVal rhif2 As integer )

Pam defnyddio paramedrau?

Mae’n gwneud y rhaglen yn haws i’w **deall**, ei **darllen**, ei **datblygu** a’i **gwirio**. Wrth ddefnyddio paramedrau mae’n haws i weld pa ddata, o du allan y modwl, sydd angen ar y modwl, er mwyn iddo gyflawni’r dasg.

Mae’n helpu osgoi newid gwerthoedd newidynnau eang ar ddamwain, a chreu gwallau yn y rhaglen.

(*Ar gyfer dealltwriaeth – nid oes angen gwybod yr isod ar gyfer arholiad hyd nes CP4*)

Paramedrau Gwerth (Value Parameters) - ByVal

Defnyddio copi o’r data yn y modwl. Bydd unrhyw newidiadau i’r paramedr gan y modwl yn cael ei **anwybyddu** (*ignored*) ar ôl gadael y modwl.

Paramedr Cyfeirio (Reference Parameter) – ByRef

Cyfeirio at y newidyn gwirioneddol. Bydd unrhyw newid i’r paramedr gan y modwl yn cael ei **gadw** (*kept*) ar ôl gadael y modwl.

Ymarfer 3

Edrychwch ar y darn isod o raglen.

Call Ciwb (rhif, ateb)

Print (“Rhif = “ & rhif & “ Ateb = “ & ateb)

Public Sub Ciwb (ByVal R As Integer ,ByRef A As Integer)

R = R \* R

A= R \* 6

End Sub

Pe bai’r newidynnau â’r gwerthoedd cychwynnol canlynol:

rhif = 2

ateb =0

Beth fydd yw’r allbwn?

Pe bai diffiniad yr is-reolwaith yn newid i:

Public Sub Ciwb (ByVal R As Integer ,ByVal A As Integer)

Beth fydd yr allbwn newydd?

Pe bai diffiniad yr is-reolwaith yn newid i:

Public Sub Ciwb (ByRef R As Integer ,ByRefl A As Integer)

Beth fydd yr allbwn newydd?

Modylau Amnyth (Nested Modules)

Mae’n bosibl i fodwl alw modwl arall. Gelwir hyn yn fodylau amnyth.

D.S. Modylau Safonol (Standard Modules)

Modwl sydd wedi cael ei greu yn barod i berfformio tasg sydd mor gyffredin mae’n cael ei gynnwys fel rhan o’r iaith. Mae’n cynnwys ffwythiannau mathemategol fel:

|  |  |
| --- | --- |
| Sqrt(x) | Ail isradd (square root) e.e. Sqr(4) yw 2 |
| Int(x) | Rhan gyfan o’r rhif e.e. Int (5.8) yw 5 |
| Rnd() | Haprif a gwerth >=0 , ond <1 |

Yn 4 prif **rhesymegol** y gellir eu defnyddio mewn algorithmau (rhaglenni)...

* **NID**
* **AC**
* **NEU**
* **XOR**

**Enghreifftiau:**

**Os nad yw (x = 100) yna output(message)**golygu y bydd y neges yn allbwn os yw gwerth x **yn** hafal i 100.

**Os yw (x > 19) ac (x < 50) yna output(message)**  
golygu y bydd y neges yn allbwn os yw'r ddau amodau (x > 19) **ac** (x < 50) yn wir … h.y. Os x yn gorwedd yn y 20..49 ystod (cynhwysol).

Gwir yn y Tabl **a** :

|  |  |  |
| --- | --- | --- |
| **AC** | **Wir** | **Ffug** |
| **Wir** | Wir | Ffug |
| **Ffug** | Ffug | Ffug |

**Os yw (x < 20) neu (y < 30) yna output(message)**  
golygu y bydd y neges yn allbwn os un **neu** arall (**neu'r ddau**) o'r amodau (x < 20) neu (y < 30) yn wir.

Tabl gwir **neu** :

|  |  |  |
| --- | --- | --- |
| **NEU** | **Wir** | **Ffug** |
| **Wir** | Wir | Wir |
| **Ffug** | Wir | Ffug |

**Os yw (x < 20) XOR (y < 30) yna output(message)**  
Byddai allbwn y neges os un **neu** arall (ond **nid y ddwy**) o'r amodau (x < 20) neu (y < 30) yn wir.

Gwir yn y Tabl **XOR** :

|  |  |  |
| --- | --- | --- |
| **XOR** | **Wir** | **Ffug** |
| **Wir** | Ffug | Wir |
| **Ffug** | Wir | Ffug |

**Cywasgu Data (Data Compression)**

# **Cywasgu ffeiliau**

* Mae swm y data yn cael ei leihau ac mae’r ffeil yn cael ei gwneud yn llai
* Caiff cywasgu ei ddefnyddio i arbed lle ar y ddisg
* Mae angen llawer o le storio ar gyfer delweddau ac felly efallai y bydd angen eu cywasgu.

Cywasgu yw’r broses o wneud maint ffeil yn llai. Gall hyn fod yn fanteisiol gan ei bod yn galluogi storio mwy o ddata ar y ddisg a hefyd gall ffeiliau gael eu trosglwyddo’n gyflymach. Mae dau ddull o gywasgu disg: y naill yn seiliedig ar feddalwedd a’r llall yn seiliedig ar galedwedd.

Yn aml mae cywasgu disg sy’n seiliedig ar feddalwedd yn cael ei gynnwys fel cyfleuster system weithredu ac felly mae ar gael yn barod ar y rhan fwyaf o systemau cyfrifiadurol. Anfantais hyn yw ei fod yn arafu’r broses o ddarllen ac ysgrifennu i ddisg.

Ar gyfer cywasgu disg sy’n seiliedig ar galedwedd mae angen caledwedd arbenigol, sy’n gallu bod yn ddrud. Fodd bynnag, nid yw’n effeithio ar gyflymder cyrchu gymaint â chywasgu disg sy’n seiliedig ar feddalwedd.

Mae cywasgu sy’n seiliedig ar ddisg bob amser yn ddigolled.

# **Mathau o gywasgu (gan gynnwys colledus a digolled)**

Mae dau brif ddull sy’n cael eu defnyddio i gywasgu ffeiliau sydd wedi’u storio ar system gyfrifiadurol, sef *colledus* a *digolled*.

## **Cywasgu digolled**

Mae cywasgu digolled yn defnyddio algorithm sy’n cywasgu data i ffurf a all gael ei datgywasgu yn ddiweddarach heb golli data, gan ddychwelyd y ffeil i’w hunion ffurf wreiddiol. Mae’n ddewis gwell na chywasgu colledus pan fyddai colli unrhyw fanylion, er enghraifft mewn rhaglen gyfrifiadurol neu ddogfen wedi’i gairbrosesu, yn gallu cael effaith niweidiol.

Fersiwn syml posibl o gywasgu digolled ar ddogfen wedi’i gairbrosesu yw amnewid llinyn cyffredin, fel y gair Saesneg ‘*the*’, â rhywbeth fel y symbol @. Byddwch wedi dysgu ar dudalen 21 fod un nod yn cymryd 1 beit o gof; felly, byddai’r llinyn ‘the’ yn cymryd 3 beit.

|  |  |  |
| --- | --- | --- |
| **Testun gwreiddiol heb ei gywasgu** | The word the, is the most frequently used word in the English language. | 71 nod (beit) |
| **Testun cywasgedig** | @ word @, is @ most frequently used word in @ English language. | 63 nod (beit) |

Mae hyn yn lleihad o 11% ym maint y ffeil!

### **Nodweddion**

* Gall y neges wreiddiol gael ei datgywasgu yn ôl i’w ffurf wreiddiol (mae’n adfer yr holl ddata gwreiddiol)
* Mae cywasgu data digolled yn gweithio drwy ddarganfod patrymau wedi’u hailadrodd mewn data a chywasgu’r patrymau hyn mewn ffordd effeithlon. Am y rheswm hwn, cyfeirir at gywasgu data digolled fel lleihau afreidrwydd *(redundancy*). Gan fod lleihau afreidrwydd yn dibynnu ar y patrymau yn y neges, nid yw’n gweithio’n dda ar negeseuon hap. Mae cywasgu data digolled yn ddelfrydol ar gyfer testun. Mae’r mwyafrif o’r algorithmau ar gyfer cywasgu digolled wedi’u seilio ar y dull cywasgu *LZ* a ddatblygwyd gan Lempel a Ziv
* Un math o amgodio testun sy’n effeithiol iawn ar gyfer ffeiliau â llinynnau hir o ddidau ailadroddus yw *RLE*. Mae *RLE* yn sefyll am Run Length Encoding (Amgodio Hyd Rhedeg)
* Mae *RLE* yn defnyddio dull geiriadur llithr yr algorithm *LZ*. Mae’r dull geiriadur llithr yn defnyddio pwyntyddion o fewn y ffeil gywasgedig sy’n pwyntio at linynnau o ddidau wedi’u cynrychioli’n flaenorol o fewn y ffeil
* Dyma enghraifft o neges a allai gael ei hamgodio’n effeithiol gan ddefnyddio *RLE*:
  + Y gair ‘the’ yw’r gair a ddefnyddir amlaf yn y Saesneg. Gallai’r llinyn “the” gael ei gynrychioli unwaith yn unig a gellid pwyntio ato gan bob galwad ddiweddarach i’r llinyn hwnnw
* Mae codio Huffman yn gweithio drwy ddadansoddi amlder yr elfennau mewn data. Neilltuir yr amgodio byrraf (gyda’r lleiaf o ddidau) i’r elfennau sy’n digwydd amlaf. Neilltuir yr amgodio hwy (gyda mwy o ddidau) i’r elfennau sy’n digwydd yn llai aml
* Gallai codio Huffman gael ei ddefnyddio i gywasgu ffeiliau sain, yn enwedig recordiadau sy’n cynnwys amleddau o’r math a glywir yn y llais dynol

## **Cywasgu colledus**

Mae cywasgu colledus yn dechneg cywasgu data sy’n cywasgu maint y ffeil trwy waredu rhywfaint o’r data. Bwriad y dechneg yw lleihau swm y data sydd angen eu storio.

Mae’r fersiynau gwahanol o logo CBAC isod yn dangos faint o’r data sy’n gallu cael ei waredu, a sut mae ansawdd y delweddau yn gwaethygu wrth i’r data a ffurfiodd y gwreiddiol gael eu gwaredu. Yn nodweddiadol, mae’n bosibl gwaredu swm sylweddol o ddata cyn i’r defnyddiwr sylwi ar y canlyniad. Mae’r gymhareb gywasgu yn cael ei chyfrifo gan ddefnyddio’r fformiwla syml:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Delwedd wreiddiol 100 kB | Delwedd gywasgedig 10 kB  (cymhareb gywasgu = 100/10 = 10 neu 10:1) | Delwedd gywasgedig 5 kB  (cymhareb gywasgu = 100/5 = 20 neu 20:1) |
| **Maint y ffeil** | **Maint y ffeil** | **Maint y ffeil** |

Mae cywasgu colledus yn cael ei ddefnyddio hefyd i gywasgu data amlgyfrwng, fel sain a fideo, yn enwedig mewn rhaglenni sy’n llifo cyfryngau dros y Rhyngrwyd.

### **Nodweddion**

* Ni all ffeiliau sydd wedi’u cywasgu byth gael eu hadfer i fod yn union yr un fath ag yr oeddynt cyn cywasgu
* Pan gaiff ffeiliau wedi’u cywasgu eu datgywasgu nid ydynt yn rhoi’n ôl y data gwreiddiol, h.y. mae data’n cael eu colli
* Gan nad yw cywasgu colledus yn gallu rhoi’r union ddata gwreiddiol ar ôl datgywasgu, nid yw’n ddull cywasgu da ar gyfer data allweddol, fel data testunol
* Mae’n fwyaf defnyddiol ar gyfer data analog wedi’u samplu’n ddigidol, fel sain, fideo, graffigau neu ddelweddau
* Mae algorithmau ar gyfer cywasgu colledus yn amrywio, ond mae llawer yn defnyddio blaendoriad lefel trothwy. Mae hyn yn golygu bod yr holl ddata ar ôl lefel a ddewisir yn cael eu blaendorri, e.e. mewn ffeil sain, gallai’r amleddau uchel ac isel iawn na all y glust ddynol eu clywed gael eu blaendorri o’r ffeil
* Rhai enghreifftiau o algorithmau cywasgu data colledus yw JPEG, MPEG, ac MP3