Algorithm	Id: AgnewGerry2014LabEx1	Purpose: BMI Costs	
Constants	**Type**	**Name**	**Value**
	final double	UNDER_WEIGHT	= 100.00
	final double	NORMAL_WEIGHT	= 120.00
	final double	OVER_WEIGHT	= 155.00
	final double	SEVERE_WEIGHT	= 200.00
	final double	TEESHIRT	= 5.00
	final double	DISCOUNT_FRIENDREF	= 0.02
	final double	DISCOUNT_RETURNCL	= 0.035
	final double	DISCOUNT_CORPORATE	= 0.04
	final double	VAT_RATE	= 0.23
Variables	**Type**	**Name**	**Initial Value**
	int	i, maxClientNum, months,	
	int	highClientNumber	= 0.00
	int	lowClientNumber	= 0.00
	char	typeBMI, teeShirt	
		discountType	
	double	kilos, metres, promoValue	
	double	bmi, packageCost	
	double	teeShirtCost,	
	double	discountAmt, fullCost	
	double	vatCost, finalCost	
	double		
	double	discountRate	= 0.00
	double	totPackageCost	= 0.00
	double	totDiscountAmt	= 0.00
	double	totPromoValue totVatCost	= 0.00
	double	totFullCost totFinalCost	= 0.00
	double	highPackageCost	= 0.00
	double	lowPackageCost	= 99999.99

Input	
	Number of clients?: ➔ n (for example 6)
Line Input	**Kilos/Metres/Months/TShirt/Disc/Promo/:** **56/ 1.6 / 1 / Y / F / 20.00 (repeated n times inside for)**
Output	
	Headers (Name / =======) – last
header	**Line Output (i, bmi, typeBMI, packageCost, discountAmt, promoValue, fullCost, vatCost, finalCost)**
line	
footer	**Footers (totals, largest & smallest) – last**
Expected Results	
	As per the Exam sheet screenshot

Steps	Pseudocode
1	Define top of program **Comments**: Program Id, Developer, Date & Purpose (over)
2	Define **Constants, Variables** and any required **Initialisation** (page 1)
3	**Preliminary Input:** **maxClientNum= Input ("Enter number of clients: ")**
5	**Output (headers/your name/====) // last when working**
6	**for (i = 1 to maxClientNum) // process each client's details** **{**
7	**// Line Input** **kilos/ metres / months / teeShirt /discountType /promoValue** **= Input ("Kilos/Metres/Months/TShirt/Disc/Promo/: ")**
8	**Calc BMI = kilos / (metres * metres)** **(if/else/if) for BMI type and package cost** **(bmi<18.5) ➔typeBMI='U' packageCost = UNDER_WEIGHT** **((bmi>=18.5)&&(bmi<=24.9)) ➔typeBMI='N' packageCost = NORMAL_WEIGHT** **(bmi>=25)&&(bmi<=29.9)) ➔ typeBMI='O' packageCost = OVER_WEIGHT** **otherwise ➔typeBMI='S' packageCost = SEVERE_WEIGHT**
9	**Calc packageCost *= months**
11	**Calc discountRate (if/else/if)**
12	**discountType F ➔ discountRate = DISCOUNT_FRIENDREF**
13	**discountType R ➔discountRate = DISCOUNT_RETURNCL**
14	**discountType R ➔discountRate = DISCOUNT_RETURNCL**
15	**discountType C ➔discountRate= DISCOUNT_CORPORATE;**
16	**discountType P ➔ (packageCost>450) : discountRate= 0.06, (packageCost>300) :** **discountRate= 0.04, (packageCost>=150) discountRate= 0.02, Otherwise 0**
17	

18	Calc teeShirt = teeShirtCost = TEESHIRT (if/else) otherwise 0
19	Calc discountAmt = packageCost * discountRate
20	Calc fullCost = (packageCost - discountAmt) + teeShirtCost-promoValue
19	Calc vatCost = fullCost * VAT_RATE;
20	Calc finalCost = fullCost + vatCost
21	Output (unformatted / println ( )) Output (formatted / printf ( )) // subsequently
22	// Accumulate totals – here later
23	} // for
24	Output (totals) & (largest & smallest)